Exploiting structure in man-made environments

ALPER AYDEMIR

Doctoral Thesis
Stockholm, Sweden, 2012

TRITA-CSC-A 2012:14
ISSN-1653-5723
ISRN-KTH/CSC/A–12/14-SE
ISBN 978-91-7501-549-1

KTH
SE-100 44 Stockholm
SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av doktorsexamen 20 November 2012 klockan 10:00 i Sal F3, Lindstedtsvägen 26, KTH Campus Valhallavägen, Stockholm.

Tryck: US AB

Abstract

Robots are envisioned to take on jobs that are dirty, dangerous and dull, the three D's of robotics. With this mission, robotic technology today is ubiquitous on the factory floor. However, the same level of success has not occurred when it comes to robots that operate in everyday living spaces, such as homes and offices.

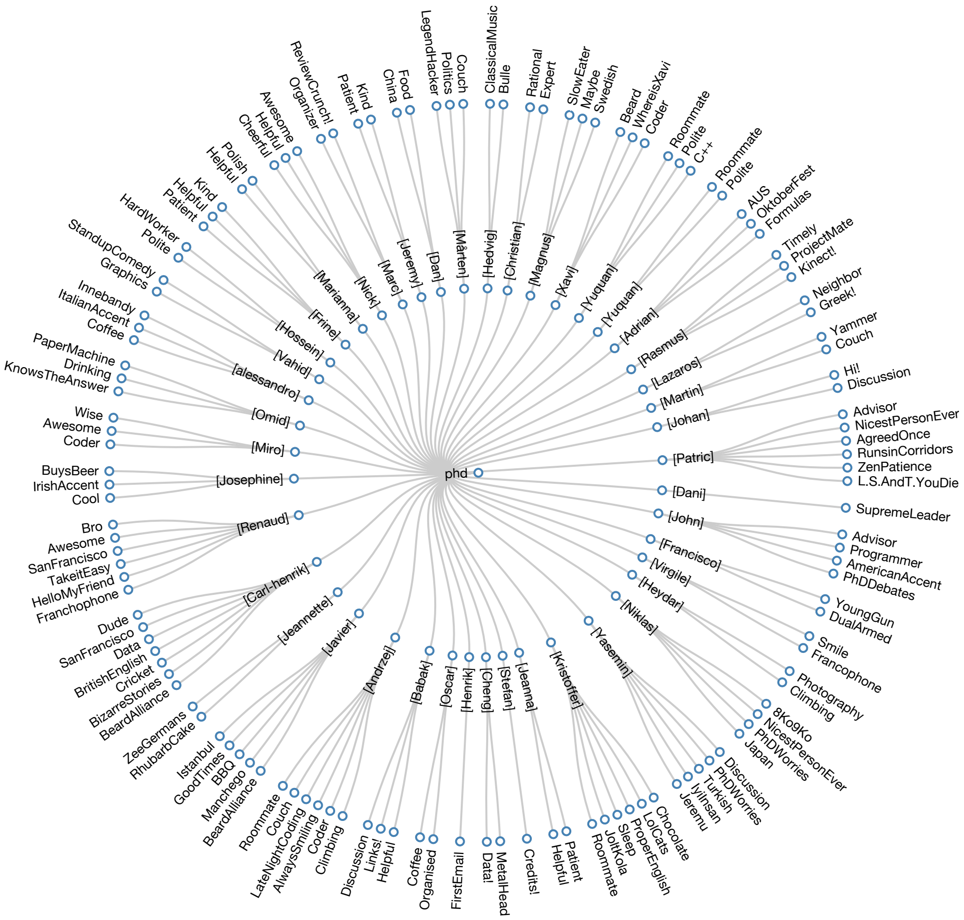
A big part of this is attributed to domestic environments being complex and unstructured as opposed to factory settings which can be set up and precisely known in advance. In this thesis we challenge the point of view which regards man-made environments as unstructured and that robots should operate without prior assumptions about the world. Instead, we argue that robots should make use of the inherent structure of everyday living spaces across various scales and applications, in the form of contextual and prior information, and that doing so can improve the performance of robotic tasks.

To investigate this premise, we start by attempting to solve a hard and realistic problem, active visual search. The particular scenario considered is that of a mobile robot tasked with finding an object on an entire unexplored building floor. We show that a search strategy which exploits the structure of indoor environments offers significant improvements on state of the art and is comparable to humans in terms of search performance. Based on the work on active visual search, we present two specific ways of making use of the structure of space. First, we propose to use the local 3D geometry as a strong indicator of objects in indoor scenes. By learning a 3D context model for various object categories, we demonstrate a method that can reliably predict the location of objects. Second, we turn our attention to predicting what lies in the unexplored part of the environment at the scale of rooms and building floors. By analyzing a large dataset, we propose that indoor environments can be thought of as being composed out of frequently occurring functional subparts. Utilizing these, we present a method that can make informed predictions about the unknown part of a given indoor environment.

The ideas presented in this thesis explore various sides of the same idea: modeling and exploiting the structure inherent in indoor environments for the sake of improving robot's performance on various applications. We believe that in addition to contributing some answers, the work presented in this thesis will generate additional, fruitful questions.

Acknowledgement: First things first, I gratefully acknowledge the Swedish Foundation for Strategic Research SSF through the Centre for Autonomous Systems (CAS) and the Framework Program 7 project "CogX" (ICT-215181).

I consider myself quite lucky for having Patric as my primary advisor. If it wasn't for him, this work would not have materialized. I'd like to thank John as well for always having time, comments and suggestions on this thesis. Thanks Kristoffer and Andrzej for the fruitful discussions and making all-nighters more pleasant at the lab. Special thanks to Friné for always being patient with me. Thanks to everyone at CVAP and CogX for being awesome colleagues throughout. It's impossible to sum up years of experience and interaction with some 50+ people here. I'm pretty sure they already know how awesome I think they are. So I made this instead.



Contents

Anne Babama ve Burcu'ya...

Chapter 1

Introduction

"The baby, assailed by eyes, ears, nose, skin, and entrails at once, feels it all as one great blooming, buzzing confusion..."
William James, Principles of Psychology, 1890

In this chapter we will set the stage for the rest of the ideas that will be presented henceforth in this thesis by arguing that intelligent agents need to make heavy use of the inherent spatial structure of the environment.

As robotics shifted its focus from factory floors to everyday living spaces, researchers regarded the new, uncharted domestic environments domain as *unstructured.* This meant that we needed to equip our robots with a wide range of ever more capable sensors to be able to make sense out of the confusing buzz of the outside world. However, with sensors came uncertainty and with uncertainty came the need to deal with untrustworthy streams of percepts from various sensors. As a result, researchers motivated a long list of methods and algorithms that could handle the chaos of our everyday living spaces as opposed to the precise status of the manufacturing floor. This shift in thinking manifested itself in inclining towards probabilistic methods which could deal with this uncertainty in a much better way than classical AI methods. Breakthrough results have allowed robots to achieve various capabilities, simultaneous localization and mapping being one.

However as a by product of this line of thinking, also came the notion that *assumptions and priors are things to be avoided.* The thinking went: "If the world is unstructured and highly unpredictable then our best bet is to rely on no assumptions about the world at all." [1]. This line of thinking is valid as long as the precondition for it (that the world is unstructured), holds true. It does not. In fact, we would argue that our everyday world is highly structured. Homes are organized into rooms and areas, where each room has a narrow range of uses (kitchen, bedroom etc.) which has some basic furniture that is largely immobile. Same can be

[1] A web search with the keyword unstructured and robotics results in thousands of publications, ranging from humanoids operating in kitchens to mobile robots navigating indoors.

said of offices, hospitals and most other human environments. Even objects that can easily be moved such as cups or books spend most of their time in specific places and are not randomly distributed all over the environment. As humans, we heavily make use of this structure in our everyday life. It follows therefore that robotic systems should also be able to exploit such regularities in spatial structure in carrying out various tasks.

Often, in robotics research it is stated that factory floors can be designed and known to the minute detail whereas places outside the factory floor are messy, unpredictable and without apparent structure. In the former case, industrial robots have enjoyed formidable advantages from knowing beforehand how the world is configured and will behave, industrial robots are now an essential part of manufacturing. The same level of success does not apply so far when it comes to service or domestic robotics or even to industrial robots in less strict settings. The lion's share for this failure is typically attributed to domestic environments being complex and unpredictable. As this may be the case, we believe part of the reason for robots not thriving as much in man-made environments is ignoring spatial structure and attempting to rely on the least amount of assumptions about the world. Therefore robotic systems are deprived from valuable and correct prior spatial information which would be of great help in a variety of tasks. It is then our claim that when building intelligent robots, the inherent spatial structure is an information source to be heavily exploited.

Before one can go about realizing this idea, several questions need to be answered. For most real-world complex tasks, commonly the most successful way of obtaining prior information is simply collecting data about the problem and analyzing it. However we see the following as noteworthy challenges:

1. Deciding on what type of priors and data is relevant to the task
2. How to collect the data
3. How to model the prior information based on this data
4. How to utilize and maintain the priors extracted from this data

The first point is often ignored and hardly discussed in the literature. That is, for a given robotic capability, what should be the nature of priors that the system should rely on? This requires a deep understanding of the task at hand, as the subtleties in the way of accomplishing the task are often invisible from the surface. A good way of gaining insights into a task is by simply attempting to build a system and uncovering bits and pieces that constitutes the most crucial challenges along the way. We will do just that in the next chapter.

Second, depending on the type and the scale, data collection is often a hard logistics problem. If we base our solution on learning from data, then the solution's quality highly depends on the data characteristics. It is indeed possible to shield ourselves from this aspect of the problem by building admittedly limited datasets and emphasizing the models learned or the idea behind the approach. However, we feel this won't address the main problem as it only delays it. There is a large body of research in how to ease the process of collecting large amounts of quality data [1, 2], which is often outside the scope of works in robotics, though can be very useful. Through the work which resulted in this work, we have also encountered this problem a number of times and the solutions we have come up with are part of this thesis.

The third point entails capturing what is in the data accurately. Amongst the stated points, the bulk of research focus on this area and rightly so, it is crucial to make good sense of the data which is enters the domain of statistics, machine learning and increasingly so combined with probabilistic reasoning techniques.

Fourth and finally, the way that the prior information is being put to use throughout the system is crucial for successful task completion. An important topic here is how to update the priors over the lifespan of a robotic system. This and the so called life long learning is an emerging area which has lots of open questions yet to be answered.

The exploitation of spatial structure in robotics is an idea with recently increasing interest and thus as stated above most of the hard problems are hidden from thought experiments. A sure way of uncovering these hard problems is simply by picking a suitable application and trying to build it. We have chosen a task that is highly challenging and involving all of the above points, active visual search (AVS) in large unknown environments, a yet unsolved problem in robotics.

In an AVS scenario, the robot is tasked with finding a known object in the environment with a camera. In the most general sense, the robot does not know the environment beforehand, the only clues it receives are from its own sensors. Since we are considering large spaces of the scale of whole buildings, that are open to exploration, the robot is required to tackle visual search in different spatial scales, first deciding where to search in the larger map-level scale and then locating the object in single scenes. The robot needs to be able to *plan* a search strategy, one that is cost efficient and most likely to lead to succeed. In the next chapter we will analyze the cases where the prior information is either absent or present and its effect on the search performance.

1.1 Outline

The outline of this thesis is as follows. In Chapter 2, we explain how to construct an active visual search system from various aspects and present our solution. As a result of this work, we identify improvement areas where exploiting spatial structure can make significant contribution. In Chapter 3, we propose the idea of using local 3D geometry as a strong indicator of object locations in depth images of everyday scenes. In Chapter 4, we look at a bigger scale and reason about unexplored space in entire building floors. We analyze a large dataset of real indoors floorplans and attempt to answer the question of predicting what lies ahead in the topology of indoor environments. Finally in Chapter 5, we conclude the work presented in this thesis lay out potential improvement areas.

1.2 Contribution

Parts of this thesis have been previously published as journal and conference articles. This thesis contains a subset of the research work done throughout the PhD that resulted it. Particularly, publications numbered 1, 2 and 3 constitutes Chapters 2-4 respectively. Below we explain the individual contributions of the author of this thesis for each paper.

1. Alper Aydemir, Andrzej Pronobis, and Patric Jensfelt. Active Visual Search in Unknown Environments Using Uncertain Semantics
   In *IEEE Transactions on Robotics.* Conditionally accepted (in review), October 2012. [3]

   Summary and Individual Contribution: This paper is on how to efficiently search and locate objects in unknown environments the size of an entire building floor, as opposed to previous work where either the search space is limited in size (typically ranging from a table top to a room) or where the environment is assumed to be known in advance, including objects, rooms and their categories. The idea presented in this work is to utilize a divide and conquer approach to search, by making use of a hierarchical modeling of space which is augmented by semantics such as room categories. The contribution of the author of this thesis is in devising the search strategies for efficient active visual search in large unknown environments by combining semantic mapping and efficient planning. For this work we have adapted the probabilistic semantic mapping approach presented in [4] to our problem and utilized the planning framework presented in [5]. The goal of this work was to attempt to solve a hard problem in robotics which involves a large view of how to utilize priors on the inherent structure of man-made environments. As such, Chapter 2 largely is based on this paper.

2. Alper Aydemir and Patric Jensfelt. Exploiting and modeling local 3D structure for predicting object locations
   In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Algarve, Portugal, October 2012. **Best Paper Finalist**. [6]

   Summary and Individual Contribution: In this work we have argued that the location of objects in everyday scenes is highly correlated with the 3D structure around them, which we have called the *3D context* of an object. The idea was to utilize this prior information in order to better guide various search processes aimed at finding objects (such as object detection) towards regions of the image where they are most likely to contain the object. This idea came from an observation while working on the active visual search system presented in Chapter 2 that most state-of-the-art computer vision algorithms have a hard time dealing with images taken when the robot is in motion looking for objects and the whole image is scanned for the target object. Instead we propose to use local 3D geometry as a very strong indicator on the location of objects in everyday scenes. The contributions of the author of this thesis include proposing and motivation the initial idea, coming up with a way of modeling the 3D context, devising and running experiments to show the applicability of the idea.

3. Alper Aydemir, Patric Jensfelt and John Folkesson. What can we learn from 38,000 rooms? Reasoning about unexplored space in indoor environments
   In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Algarve, Portugal, October 2012. [7]

   Summary and Individual Contribution: This paper originates from the idea that although a large portion of robotics is aimed towards building robots and algorithms that can robustly operate indoors, we have little idea on actual indoor environments analyzed at large scales. In this work, we have worked with a floor plan dataset containing 200 buildings and approximately 38,000 rooms, several orders of magnitude more than found in previous work. The specific question we wanted to investigate in this work is: "Given a partial floor plan of a building, can we predict the rest accurately?". Again, this question stems from a practical need in our work on active visual search, namely that how can a robot make informed decisions while exploring an environment with the purpose of finding an object? The author of this thesis contributed in coming up with the question, the idea and the effort of gathering a large indoor dataset, as well as the algorithms presented in this work and experiments performed.

4. Alper Aydemir, Moritz Göbelbecker, Andrzej Pronobis, Kristoffer Sjöö, and Patric Jensfelt. Plan-based object search and exploration using semantic spatial knowledge in the real world
   In *Proc. of the European Conference on Mobile Robotics (ECMR)*, Örebro, Sweden, September 2011. [8]

   Summary and Individual Contribution: This paper reports early progress on the active visual search work that is the topic of Chapter 2. The contributions of the author of this thesis are the same as [3].

5. Alper Aydemir, Kristoffer Sjöö, John Folkesson, Andrzej Pronobis and Patric Jensfelt. Search in the real world: Active visual object search based on spatial relations

In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA).* [9]

Summary and Individual Contribution: In this work we have explored the idea of using spatial relations for active visual search in order to cut down search space and acquire views that has a higher likelihood of bringing the target object into the field of view of the robot. We have utilized the spatial relations idea and implementation details presented in [10] and in [11].The contribution lies in evaluating the use of spatial relations in the context of an active visual search task. We have assumed that the metric map of the environment is known in advance. Furthermore, we have also assumed to have probabilistic prior knowledge on the spatial relations between objects (e.g. the cup is on the table with a certain probability). Using these, we have utilized a Markov Decision Process (MDP) planner in order to determine search strategies. With this work we have made inroads in utilizing higher level spatial concepts such as spatial relations to guide the search process. This paper is not included in the thesis although it is highly related to Chapter 2.

6. Kristoffer Sjöö, Alper Aydemir, David Schlyter, and Patric Jensfelt. Topological spatial relations for active visual search
   Robotics and Autonomous Systems. July 2012. [10]

   Summary and Individual Contribution: This work is an expansion on [9] and [11] where the idea is to use spatial relations *ON* and *IN* to aid in active visual search. The contribution of the author of this thesis is on how to utilize spatial relations for an active visual search task with a mobile robot.

7. Kristoffer Sjöö, Alper Aydemir, Thomas Mörwald, Kai Zhou, and Patric Jensfelt. Mechanical support as a spatial abstraction for mobile robots
   In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 2010. [11]

   Summary and Individual Contribution: This work presents solely in detail the spatial relations utilized in [9] and in [11]. The author of this thesis contributed in designing the perceptual model of the presented spatial relations *ON* and *IN* in a way that is useful to mobile robotics tasks.

8. Alper Aydemir, Daniel Henell, Patric Jensfelt and Roy Shilkrot, 2012. Kinect@Home: Crowdsourcing a Large 3D Dataset of Real Environments
   In 2012 AAAI Spring Symposium Series, Stanford University, CA, USA. [12]

   Summary and Individual Contribution: In this work we have attempted to amass a very large dataset of RGB-D images from natural man-made environments by making it easy to record Kinect videos, processing them to produce a 3D map and displaying the resulting 3D maps back to users to encourage participation. This effort is related to the question "How to collect the data" asked previously in this chapter. The author of this thesis contributed by coming up with the whole idea, designing the 3D mapping algorithm, designing the mapping pipeline and implementing parts of it. The project gathered significant interest worldwide, being reported in news organizations such as the Wired and the BBC amongst dozens of others. The effort is still on going and the RGB-D data pool keeps getting bigger and bigger.

9. Andrzej Pronobis, Kristoffer Sjöö, Alper Aydemir, Adrian N. Bishop, and Patric Jensfelt. 2009. A framework for robust cognitive spatial mapping
   In *Proc. of the 14th IEEE International Conference on Advanced Robotics (ICAR)*, Munich, Germany, June 2009. [13]

   Summary and Individual Contribution: This work presents a spatial modeling approach for mobile robots that can support variety of tasks such as finding objects and acquiring a semantic map of the environment. The contribution of the author of this thesis is in designing the framework in collaboration with other authors.

10. Marc Hanheide, Charles Gretton, Richard Dearden, Nick Hawes, Jeremy Wyatt, Andrzej Pronobis, Alper Aydemir, Moritz Göbelbecker and Hendrik Zender. Exploiting probabilistic knowledge under uncertain sensing for efficient robot behaviour
    Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011), Barcelona, Spain. [14]

    Summary and Individual Contribution: This work presents a way of using probabilistic knowledge coming from integrating common-sense knowledge about the world to robot's observations and the usage of a continual planner to select actions in the light of useful but uncertain data about the world. The specific application chose is active visual search. The contributions of the author of this thesis is to adapt the active visual search methods described in our work [3, 10] in the context of the system described in the paper.

11. Alper Aydemir, Kristoffer Sjöö, and Patric Jensfelt. 2010. Object search on a mobile robot using relational spatial information
    In *Proc. of the 11th International Conference on Intelligent Autonomous Systems (IAS)*, Ottawa, Canada, August 2010. [15]

Summary and Individual Contribution: This paper presents our early attempts at building a mobile robot that can search for objects in large environments using properties and regularities of man-made environments. The specific idea explored here is to use spatial relations. The contributions of the author of this thesis is devising and implementing search method and techniques that utilizes spatial relations.

12. Andrzej Pronobis, Kristoffer Sjöö, Alper Aydemir, Adrian N. Bishop, and Patric Jensfelt. 2010. Representing spatial knowledge in mobile cognitive systems
In *Proc. of the 11th International Conference on Intelligent Autonomous Systems (IAS)*, Ottawa, Canada, August 2010. [16]

Summary and Individual Contribution: This paper contains preliminary efforts in designing a spatial modeling framework for mobile robots in the same vein as [13].

Chapter 2

Active Visual Search

In the previous chapter we have argued that many robotics tasks would benefit from exploiting the spatial structure of everyday environments. In order to explore this idea, we have picked a yet unsolved, hard problem in robotics which can greatly benefit from using spatial structure. We have chosen the active visual search (AVS) problem, which deals with locating objects in large unknown environments with a camera mounted on a mobile robot. We believe investigating such a problem can in turn help us understand more about the open questions we have posed in the beginning of this thesis.

Much work is going into overcoming the problem of making sense of complex environments, to build maps augmented with semantics and objects, sometimes for long periods of time. Key in this effort is the apprehension of objects. Objects hold an important role in human perception of space [17]. Localizing and interacting with them lies at the heart of various robotics research challenges, and while there is no shortage of open questions in dealing with objects, the bulk of previous work relies on the assumption that the particular object in question is already within the sensory reach of the robot. An often stated reason for this is that tasks such as object recognition and object manipulation are already challenging enough. Nevertheless, as the field advances in its aim to build versatile service robots, the assumption of objects being readily available in the field of view of a robots sensors is no longer reasonable. Furthermore, most of the state-of-the-art solutions to the AVS problem suffers from the ”no-assumptions about the world” mindset described earlier, a point which will be more clear when we examine previous work thoroughly. For these reasons, the process of attempting to solve a realistic AVS scenario with a concrete implementation can lead us to discover various ways in which the questions posed in Chapter 1 can be addressed.

Next, we will give an example AVS scenario and formalize the problem. We will review the earliest attempts as well as the most promising AVS solutions in the recent literature from various aspects. This recap of previous work will show us that there is a great opportunity for extending the state-of-the-art AVS methods by devising search strategies which make use of environment semantics. We will lay out exactly what parts of environment semantics are useful for an object searching robot and propose several ideas on better ways to search for objects by incorporating them into an object search system. Then we will implement those ideas for a mobile robot equipped with several sensors. Experiments for applications such as AVS where a robot has to *actively* decide on the next action often tend to be tedious and long running. Furthermore, the evaluation criteria and ground truth for such applications is non-trivial. Imagine the case where a system is required to detect objects in a pre-recorded set of images versus where a robot is required to locate a stapler in an entire building floor. While in the former case the success of the system is clearly depending on the whether or not the object is detected, in the latter case we must ask the question: What is *the ideal* object search run? For this reason and in order to subject our implementation to a fair and through evaluation, we will report various kinds of experiments to the demonstrate the applicability and effectiveness of the proposed ideas.

Finally, as per usual, when attempting to solve a hard problem we will discover new gaps in the current state of research in the direction of the topic of this thesis – extracting and exploiting structure in human environments – which will push us into exploring new ideas, making the up the rest of the chapters in this thesis.

2.1 Introduction

An AVS task is ultimately related to the spatial properties of the world. Imagine a scenario depicted in Figure 2.1 in which a mobile courier robot is tasked with finding and fetching an object, located somewhere on an unknown office floor. With the limited field of view of robotic sensors, it is unreasonable to assume that the robot will exhaustively examine the whole space in order to locate the object since it requires capturing and analyzing millions of images, rendering the system unusable in practical applications. Certain types of objects are likely to be at certain locations and not distributed randomly in the world. As illustrated in Figure 2.1, food related objects are likely to be clustered in the kitchen area, John's coffee cup mostly frequents his office, kitchen and the meeting room[1], while a stapler often is in any office room or printer area.

In order to make use of such regularities, semantic information about the object and the environment can be obtained and used to derive a more efficient strategy. As an example, if the robot is looking for a coffee cup, perhaps first looking in the kitchen would result in higher success and efficiency as opposed to looking randomly everywhere. However for this to happen, the robot must first plan to find a kitchen, then efficiently find one, and then efficiently search the found kitchen. If one of these steps fail, e.g. there's no kitchen, then the robot should suitable search

[1] We should also note the difference between an *instance* of an object, such as John's cup versus any cup. This distinction has implications on the appearance and location of objects, important aspects for a mobile robot.

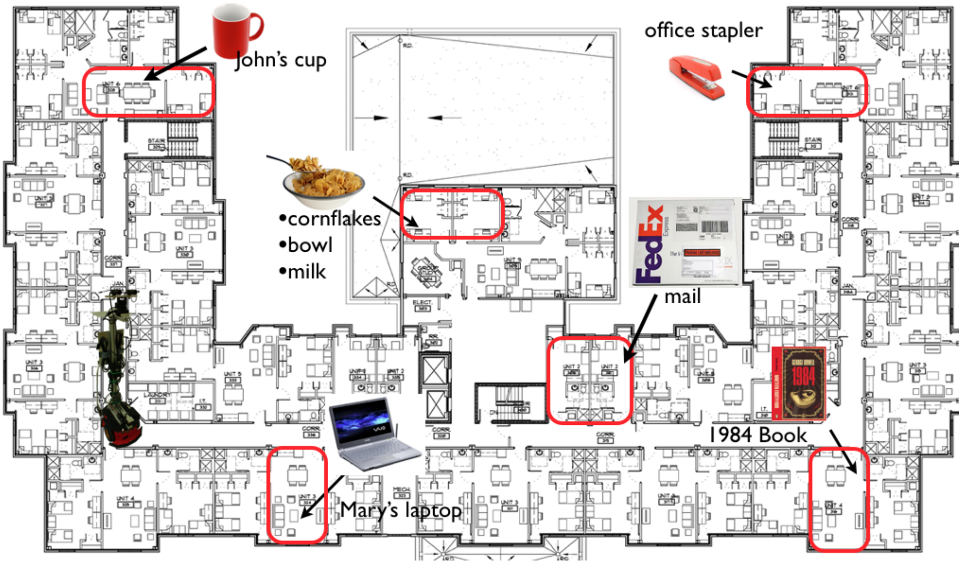


Figure 2.1: The object search scenario investigated is concerned with finding objects in large scale environments that are unknown to the robot at the start of the search.

strategies instead of giving up entirely. Therefore, a robot equipped with general world knowledge about space and objects can greatly outperform one that is not so.

In most realistic scenarios where the environment is at best partially known[2], the robot needs the ability to constantly acquire knowledge about the environment in which it operates autonomously. This adds another level of complexity to the problem of active visual search where not only direct cues with regard to the object should be used (e.g. visual feature matches in acquired images), but also *indirect* cues which might increase the odds of finding the object should be taken into account. As we discussed earlier, the semantics of the environment such as room categories can be used in order to improve the search strategy. However, knowledge acquisition (such as determining objects, room categories, the metric layout of the environment) during an AVS task in unexplored environments introduces an additional problem. Since the acquired knowledge can change the course of the search strategy, at any point during the search process the robot should be able to decide between exploring the environment (in order to discover additional spatial semantics or more places to search) and searching a part of space that is already explored. This is also known as the exploitation vs. exploration problem. We

[2] Note that even if the robot is presented with a complete map of the environment with objects and other semantic entities, a living space changes almost continually, making any given map inconsistent quickly over time.

assume a realistic scenario in which the robot is tasked with finding an object in a large-scale environment as a typical office environment consisting of 16 offices, a kitchen, a meeting room and a corridor, constituting a total search space of 33m×12m is presented. We measure the search efficiency as the total search time. The robot has no previous knowledge about the specific environment it is in and instead relies on a semantic prior about generic indoor spaces.

2.2 Problem Formulation

The problem of active search addressed in this chapter is that of finding an efficient strategy for localizing a certain object in a large scale, unknown, 3D indoor environment which we will refer to as Ψ following [18]. Concretely, we look for a strategy that decides what sequence of actions to execute so as to localize the object of interest while minimizing the total cost, where cost is defined as time. The robot can execute motion actions and sensing actions in the space of Ψ. The sensing actions are characterized by the pose of the robot, camera parameters and recognition algorithm.

Additionally, let $P_\Psi(X)$ be the probability distribution for the position of the target, X, in the search space Ψ. Depending on the level of a priori knowledge of Ψ and $P_\Psi(X)$ there are three extreme cases of the active search problem.

1. If both Ψ and $P_\Psi(X)$ are fully known, the problem is that of sensor placement and coverage maximization (assuming no uncertainty in the built map and probability distribution) given limited field of view and cost constraints.

2. If $P_\Psi(X)$ is unknown, but Ψ is known (i.e. acquired a priori through a separate mapping step), the agent either should utilize a generic probability distribution (such as a uniform one across the whole Ψ) or needs to gather information about the environment similar to the above case. However this exploration is for learning about the target specific characteristics of the environment.

3. If both Ψ and $P_\Psi(X)$ are unknown, the agent needs the ability to explore. The agent needs to select which parts of the environment to explore first depending on the target properties. Furthermore the agent needs to trade-off between executing a sensing action and exploration at any given point (i.e. should the robot search for the target in the partially known Ψ or explore further). This is classically known as the exploration vs. exploitation problem.

In this chapter, the third case is considered, where Ψ and $P_\Psi(X)$ are both unknown and the robot is required to explore the environment while searching. We provide the robot with common-sense knowledge, which is not environment specific and encodes relationships between high-level human concepts and functions of space. Typically, the common-sense knowledge encodes correspondences between objects, landmarks, other properties of space and semantic room categories. Such information is valuable in limiting the search space and helps humans efficiently search in unknown environments. Our goal is to also leverage this to achieve similar, efficient, behavior in artificial systems.

2.3 Related Work

Despite the recent interest in the problem of active search for objects, there are no extensive surveys in the literature on this topic. For this reason, we start with a comprehensive treatment of the early and current work.

In a seminal paper, Bajcsy introduced the term active perception [19]. The motivation for employing an active perception strategy is that perceptual processes often *seek* the desired percepts. In the author's words: "We do not just see, we look". In a system that employs active perception, sensors such as a camera can be used actively by adjusting its various parameters: zoom factor, depth of field, position and orientation in the 3D world.

Although the selection of parameters for a sensor may come across simply as a basic control problem, Bajcsy makes two major points for how active perception tasks differentiates themselves:

- The sensory information is often highly complex, rich in meaning and can be interpreted. Extracting certain features that may or may not depend on each other warrants the need for deep analysis of the input data.
- Prior knowledge plays a crucial role in the interpretation of this complex input data stream. This knowledge may come as models that are readily available or learned over time.

Building upon Bajcsy's introduction of active perception, Tsotsos more specifically considered active visual search [20]. Some of the advantages of an active strategy discussed in [20] are robustness to occlusions, possible increase of resolution and use of motion to disambiguate vision-related aspects of the world such as varying illumination conditions. Tsotsos and Ye analyzed the computational complexity of the active visual search problem and found it to be NP-Complete [20, 21]. A significant lesson from this analysis is that, active visual search strategies are more efficient than their passive equivalents. However, the increase in efficiency requires a more complex search process. Active search strategies often require prior information to direct the sensing. A planning approach that makes use of a prior on the target location together with the current world state to select the next action is part of most active search systems. Realizing this interplay between sensing and acting is far from trivial as the following points need to be addressed:

- How to build a prior for the active visual search task that reflects the state of the world?
- What are the search actions that constitutes a plausible and efficient search plan?

These design questions are of great importance to the performance of the system. We will show how a prior can be modeled, computed and utilized by an autonomous robot searching for an object in an unknown world.

Research focusing on the computation of the aforementioned prior appeared in the literature as early as 1976. Garvey presented an implementation of a vision system capable of finding objects in scenes by making use of certain assumptions about the semantic scene structure [22]. One example of a search run is given where the target object is a telephone. The system realizes that due to the small size of the target object, searching the whole image would be wasteful. Instead the system plans to *search for a table first* and then searches in the image region that corresponds to the table top for the telephone. The usage of prior knowledge discussed at length by Bajcsy and Tsotsos and realized in this system permits efficient reduction of the search space. Garvey calls this type of search *indirect search*. Wixson et al. provides quantitative results by comparing two search strategies, with and without indirect search for the same task [23]. Their findings indicate that indirect search greatly improves search efficiency.

The role of priors in humans when performing visual search has been investigated by Biederman et al. [24]. The paper describes an experiment in which the participants are required to search for a named object in different scenes. In some scenes the target object appeared to violate one or more of the following assumptions:

1. Support assumption: objects do not appear to float in scenes

2. Interposition assumption: foreground objects occlude the background

3. Probability assumption: objects do not appear in unlikely scenes, e.g. a car in a kitchen scene.

4. Position assumption: objects appear in certain positions in scenes, e.g. a car on a roadway.

5. Size assumption: objects have known average sizes.

In order to quantify the effect of these assumptions on human visual perception, 247 images of everyday scenes were shown to 42 participants. After each scene, the participants were asked to report if the target object was present or not. The objective of this experiment was to investigate if violation of world prior knowledge results in degradation of performance on the task [25]. The results indicated that when the target object violated one of the assumptions, the false negative rate increased to almost 60% from the base line rate of 23% for scenes with no violation.

Relating Garvey's vision system to Biederman's experiments, we can say that planning to search for a telephone by looking for a table first makes use of the support, probability and size assumptions. Telephones do not float in the air, the are likely to be found on tables and are small and hard to detect at a distance.

In a series of papers by Ye and colleagues, the first approaches to AVS are introduced as computing the next best view to move the camera to [26, 27]. A probability distribution over the 3D space ($P_{\Psi}(X)$ introduced in problem formulation) is assumed to be given and is tessellated into identically sized cells. Each cell contains the occupancy information as a binary state and the probability of the center of the target object being in this cell. Knowing the field of view of the camera, the probability mass covered by each view can be computed by summing over the probabilities of cells that are located in the field of view. This probability sum can be an measure of how good a certain view is for locating the searched object. In order to pick the next best view, a number of candidate view positions are sampled from the free space and the system greedily picks the view which has the highest probability sum. In the telephone search example, after finding the table, the view with the highest probability would include the table-top. The quality of the selected views clearly depends on the spatial probability distribution. The recent state-of-the-art visual search system from the same authors, [28] employs a similar strategy to object search using the humanoid robot ASIMO in a 4m×4m×1.5m search space. In parallel, [29] uses a probability map to guide the search and determine where to move, in a similar fashion to [26]. The authors present a SIFT-based method to find and estimate the 6DoF pose of a target object.

[30] presents an object detection method that can be used to compute likely positions for a given object in an image. The method is based on receptive field co-occurrence histograms (RFCH), which combines several descriptor responses into a histogram representation. The authors use this as a first step in analyzing an image with the result of a few points in the image where the object is likely to be at. Then the system zooms into these areas and searches at a finer scale. The authors present a mobile robot system for searching for objects in multiple rooms of an office floor. However the map and the location to search from are known a priori as in [31].

Similarly to [30] the idea of first finding object hypotheses with a fast visual algorithm and then zooming into likely object locations to perform more expensive computation is revisited by [32]. The object search task is divided into three separate sequential steps. First, the mobile robot system explores the environment to build an occupancy map. In the second step, the robot attempts to cover the environment as much as possible, this time with its peripheral cameras. During this step object hypotheses are computed based on depth from stereo and spectral residual saliency described in [33].

The method described in [31] utilizes object-object co-occurrence probabilities as a way to shape the prior on the object location over the search space. The map of the environment is fully known a priori. The system then plans a path in the map, that once traversed by the robot, has a high probability of spotting the object. The sequence of images while the robot is traveling along this path is analyzed to find the target object. The system is evaluated with 3 objects: chair, bicycle and monitor. The biggest limitation of the system is the assumption of a known map and previously detected objects scattered throughout the whole environment.

More recently, Velez et al. [34] presents a method that models the correlations between observations as opposed to ambitiously attempting to model the entire environment with its semantics. The observation model also takes into account the movement cost, an often neglected aspect of active perception. Furthermore, in contrast to most previous work, the subsequent observations are not assumed to be independent. This allows the robot to move to poses where the detection results will benefit the most according to the observation model. The authors present experimental runs in simulation and on a real robot with promising results that shows a clear advantage of employing active perception.

The authors of [35] have shown a similar system in which a method for place labeling is used to bootstrap the search. As [31] this approach also uses the semantics of the environment to make the search more efficient. Simulation experiments of search indicate that making use of the environment semantics results in fewer analyzed views compared to an uninformed coverage based search strategy.

The above methods provide different ways of constructing priors with various assumptions about the initial state of the robot and the environment. As stated earlier, another important point of an active perception system is the need to *plan* what sensing or moving actions are required to achieve the task. This is generally called *view planning* [36], requiring constant monitoring of the world and re-planning if necessary. We will now focus on the literature that deals with this aspect of the visual object search problem.

In its simplest form we can think of the view planning problem as covering a certain search space with sensors that have limited field of view. Often, minimizing the number of sensing actions and movement cost is desirable for increased search efficiency. Art gallery algorithms deal with this exact problem: Given a 2D polygon representing an art gallery (the search environment) and a limited amount of guards (viewpoints from which part of the environment is visible) to protect the artworks, what is the best way to place guards so as to cover the polygon fully? This problem has been extensively researched in the computational geometry literature [37]. An extensive introduction to the topic and surveys of the results can be found in [38, 37] and more recently in [39] for mobile guards.

A number of researchers adopted the algorithms from the art gallery literature to mobile robotics. [40, 41] present a randomized art gallery algorithm for mobile robots that are tasked with covering an environment. [42] presents a heuristics based method to find an object in a 2D polygon world. In a follow up work the authors present a sampling based algorithm similar to [40] to find an object in a 3D environment [43]. Such coverage based solutions provide an accurate description of the problem when the sensing capabilities of the robot are deemed noise-free and the world state is assumed to be completely known.

In a typical robotics scenario, there are uncertainties associated with sensing and action. Therefore, some recent papers tackled the problem by drawing inspiration from the planning literature. Hollinger et al. applies a POMDP planner to the problem of object search with single or multiple searchers [44]. In order to model the object search problem as a POMDP, the continuous 3D search space needs to be discretized carefully. This is due to the high computational complexity of most state-of-the-art POMDP solvers. As an example, a relatively small environment of dimensions 10m×10m×3m tessellated into 10cm sized cubes would result in $3 \cdot 10^5$ dimensional belief states which would pose a serious challenge for most planning algorithms. For this reason, the authors discretize an entire simulated office building floor into 69 rooms as possible object locations. They make the assumption that whenever the robot and the object are in the same room, the object is detected. This is a big simplification of detecting an object with a camera since the task of finding an object in a place as big as a room involves many difficulties such as calculating a good viewing position, dealing with occlusions and detecting objects that appear small in the image. The authors provide simulation results and a proof-of-concept implementation where a mobile robot is tasked to find cups in an already known environment with known search positions that the robot may choose to stop and take a picture from.

Similar to [30], [45] presents a approach where a mobile robot attempts to detect as many objects as possible in an environment of known size but with unknown obstacles. The system uses SIFT features to detect object candidates and then employs what the authors call a verification planning algorithm to confirm the presence of objects for these candidates. Further, [46] presents early results on modeling the search problem as a constrained Markov decision process (MDP). The planning problem is constrained in the sense that the authors allow a certain amount of time during which the robot has to detect as many objects as possible. The results, shown in simulation, indicate the plausibility of the approach.

Recent work in [47] and [48] investigate the usage of RFID sensors for the object search problem. Although visual search poses challenges such as illumination and viewpoint changes and object detection that RFID sensors do not suffer from, RFID antennas also have limited field of view. The system presented in [47] searches for certain product shelves in a supermarket setting. The environment is represented as a connectivity graph. The method exploits the default knowledge about supermarkets in that related products are stored in nearby shelves. The authors compare their results to human search performance measured in path length during search. [48] coins the term RF vision for building and analyzing images of the environment where each pixel represents the signal strength of a certain RFID tag in the corresponding direction. This image is used to infer the 3D location of the target object by the fusion of three sensory modalities: an RF antenna, a low resolution camera and a tilting laser scanner. The authors describe a method to fetch an object tagged with an RFID tag from an signal strength image of the scene.

The idea of extracting background knowledge on objects and human living spaces from existing data resources such as the Internet in order to find objects in large environments has recently seen interest in a series of papers [49, 50]. The authors describe a method, ObjectEval, that combines human supervision and learned background knowledge in order to compute a utility function for the object search task. This is in line with the author's earlier work on applications such as understanding natural route descriptions [51] and grounding spatial symbols in sensory data [52], all relevant competences for a mobile robot that can search for objects.

The authors in [53] present an object search system which utilizes background knowledge about typical object locations in indoor environments. The search locations in the map are assumed to be known in advance and the robot picks the order to visit these locations to find the object.

			[28]	[29]	[30]	[32]	[31]	[35]	[53]	[43]	[44]	[45]	[47]	[48]	this work
Scenario		Large-scale space	-	-	✓	-	✓	-	✓	-	-	-	✓	-	✓
		Realistic real-world environment	-	✓	-	-	✓	-	✓	✓	✓	✓	✓	✓	✓
		Quant. eval. of search method	✓	✓	-	-	✓	✓	-	-	-	-	✓	✓	✓
Knowledge Rep.	Prior W. State	Environment map	●	●	○	●	○	○	○	○	○	●	●	●	●
		Object information	○	○	○	○	●	○	●	○	○	○	●	○	●
		Place information	●	●	○	●	○	●	○	○	○	○	●	○	●
		Object-object relation	-	-	-	-	✓	-	-	-	-	-	✓	✓	✓
		Object-place relation	-	-	-	-	-	✓	✓	-	-	-	✓	-	✓
		Place-place relation	-	-	-	-	-	-	-	-	-	-	-	-	✓
Actions / Planning		Automatic viewpoint estimation	✓	✓	✓	✓	-	✓	-	✓	-	✓	-	-	✓
		Planning multiple steps ahead	-	-	-	-	✓	-	-	-	✓	✓	✓	-	✓
		Optimal plan (POMDP)	-	-	-	-	-	-	-	-	✓	-	-	-	-
		Autonomous exploration	✓	✓	-	✓	-	-	-	-	-	✓	✓	-	✓
		Exploration vs exploitation	-	-	-	-	-	-	-	-	-	✓	✓	-	✓
		Goal-directed exploration	-	-	-	-	-	-	-	-	-	✓	✓	-	✓

Table 2.1: Table comparing approaches to object search with mobile robots. Legend: ○- Given before the search started, ●- Acquired during the search process.

2.4 Analysis of the Problem

We can think of the active search problem as a decision process with a goal state and a set of actions that can take the robot from the current to the goal state. Since our observations are inaccurate and stochastic, one can formulate the problem as a Partially Observable Markov Decision Process (POMDP) [54]. In a POMDP, a probability distribution over the set of possible world states is modeled instead of directly representing the true state since the latter is not directly observable. This is called a belief state. The solution to a POMDP is a policy which specifies the optimal action at any belief state. The optimality comes at the price of computational complexity since the dimensionality of the POMDP belief state space is equal to the number of possible world states.

Let's first analyze the problem as in [28], assuming a fully explored search environment and overlaying a 3D grid on the entire map, each grid cell holding the probability of the target being there. In that case, the number of states is equal to the number of target positions in the 3D grid. As part of the POMDP formulation, we define one action which is moving the camera to a certain position and orientation and performing sensing and recognition in a single cell. The observations correspond to the outcome of the recognition algorithm, i.e. presence or absence of the target.

This leads to a computationally challenging formulation. The environment in the experimental evaluation in this thesis is 33 m × 12 m with roughly 3 m from floor to ceiling. Even with a large cell size of e.g. 0.1 m cube, this results in $1.2 \cdot 10^6$ cells. As discussed in the context of object search in [44], most general POMDP solvers can handle number of states in the order of thousands, i.e., several orders of magnitude lower. Additionally, such an approach requires a perfectly consistent 3D mapping framework and knowing the full extent of the world. Relaxing the fully explored world assumption and searching in a partially explored environment necessitates a new exploration action in addition to the search action. Deciding when to search and explore and reasoning about the outcome of an exploration action adds to the computational complexity.

A naive cell-by-cell search strategy would be extremely time consuming. A common way to reduce the search space when searching for objects is to limit the search to only occupied regions in space. In [28], the search space is limited to areas around a known table and shelf, while in [55] and more recently in [27] only regions of space where a laser scanner detects obstacles are used. In our example, such a method would reduce the number of cells from $1.2 \cdot 10^6$ to $8 \cdot 10^4$. Assuming that the camera has a 45° opening angle and it needs to be located no more than 2 meters from the object for reliable detection, $3 \cdot 10^4$ views are required to cover the space. This corresponds to approximately 12 hours of search, assuming that each view (including motion of the robot) takes 5 seconds and the object is found half way through the search. This is prohibitively slow for most realistic applications.

In order to make the search practical, we must find an heuristic that guides the search more efficiently than only using obstacles. We can get inspiration by analyzing human behavior. In a specific environment and when looking for a specific object, we would rely on detailed instance models, e.g. Patric's mug is likely to be located on Patric's desk. A robot assistant could gather similar statistics over time. In this work we assume an unknown environment. Therefore, we cannot use any specific knowledge about the objects therein. However, most humans tasked with finding a mug in an unknown environment would still not use exhaustive search. We would make use of very strong, domain unspecific priors for the location likely to contain the object. For example, we know from experience that there is a strong correlation between mugs and kitchens. Instead of looking for the mug exhaustively in the floor of a building, we would first search for a kitchen. This can be generalized to exploiting spatial correlations between object categories and room categories. We argue that efficient search in human environments should make use of such knowledge, as in [31, 35].

Finally, it is important to keep in mind that we consider exploration of unknown space as part of the problem. That is, we want to find an object without knowing the entire extent of the environment being searched. This requires a principled way of trading exploration of the unknown against search of what is already known. In order to exploit semantic information, the system needs to be able to reason, not only about the semantic spatial concepts associated with objects in the already explored part of the environment, but also about what might lay ahead.

In the next sections, we will present the design of our active search system based on this analysis, first focusing on search space modeling and pruning, and then on actions and control.

2.5 Modeling space

As pointed out above, the ability to reduce the search space is crucial for practical applications. We choose to deal with this problem by directing the search towards locations that are likely to contain the object.

Indoor environments are typically organized into rooms, each fulfilling a specific function of everyday life. At the same time, the category of a room is often strongly correlated with the actions afforded by the objects found therein (e.g. a book is more likely to be found in offices rather than in kitchens). We argue that rooms are an important spatial concept for efficiently pruning large amounts of search space in typical indoor environments[3]. Our idea is to exploit the correlation between room category and objects as part of the semantics of the environment. Rooms have frequently been used in the past as nodes in topological representations [57, 58, 59]. Here we make use of rooms as a means to implement a divide-and-conquer strategy for the object search. Once a decision to search a room is made, the system can then analyze the room through a more detailed search, involving view planning by calculating where exactly to move the camera in this smaller part of the search

[3] We note that rooms as defined here do not have to have physical boundaries such as walls and doors as demonstrated in [56]

space. Our assumption, which will be confirmed by the experimental evaluation, is that the cost of classification of rooms is more than compensated by the benefits.

Since we assume no initial knowledge of the specific environment in which the robot operates, the categories of rooms found in the environment have to be inferred based on observations acquired by the robot during the search. As we will explain in the next section in more detail, this allows us to reason about object presence in the known and unknown parts of space, by combining different types of observations (e.g. of objects and room appearance) and predicting existence of rooms of certain categories even in unexplored space.

Modeling the Search Space on the Environment Scale

Our modeling of the search space is as follows. On the large scale (e.g. a whole building floor containing multiple rooms), we represent the search space as an undirected graph called the *place map*. The nodes of the graph correspond to discrete *places* in the environment and are created at equal intervals as the robot moves. Edges in the graph represent direct paths between places. Together, places and paths represent the topology of the environment. An example of a place map is shown in Figure 2.5.

The places in the place map are further grouped into rooms by detecting doors in the environment. In addition, unexplored parts of the environment are represented in the place map using hypothetical places called *placeholders* defined in the boundary between free and unknown space in the metric map [60, 61]. Both places and placeholders are associated with beliefs about room categories estimated based on the available knowledge about the explored part of the environment. To assist in deciding which room to search or which placeholder to explore, we estimate two probability distributions related to object presence in the already discovered rooms and in unexplored space:

- $p(O^{o_i}_{r_j}|\boldsymbol{\theta})$, $O^{o_i}_{s,r_j}{\in}\{0,1\}$ - distribution indicating whether an object of the category o_i exists in not yet searched area of one of the known rooms r_j, derived from all the observations $\boldsymbol{\theta}$ collected by the robot up to this point.
- $p(O^{o_i}_{h_j}|\boldsymbol{\theta})$, $O^{o_i}_{s,h_j}{\in}\{0,1\}$ - distribution indicating whether an object of the category o_i exists in a potential new room which can be discovered after exploring in the direction of placeholder h_j, derived from all the observations $\boldsymbol{\theta}$ collected by the robot up to this point.

As noted previously, in order to calculate the above, we exploit the relationship between the room category and object presence of a certain category. Therefore, we calculate two types of room category probabilities, for explored and yet unexplored space:

- $p(C_{r_j}|\boldsymbol{\theta})$, $C_{r_j}{\in}\{c_k\}_{k=1}^{N_C}$ - distribution over room categories (for N_C categories in total) for a given known room r_j and all the observations $\boldsymbol{\theta}$ that the robot gathered up to this point.

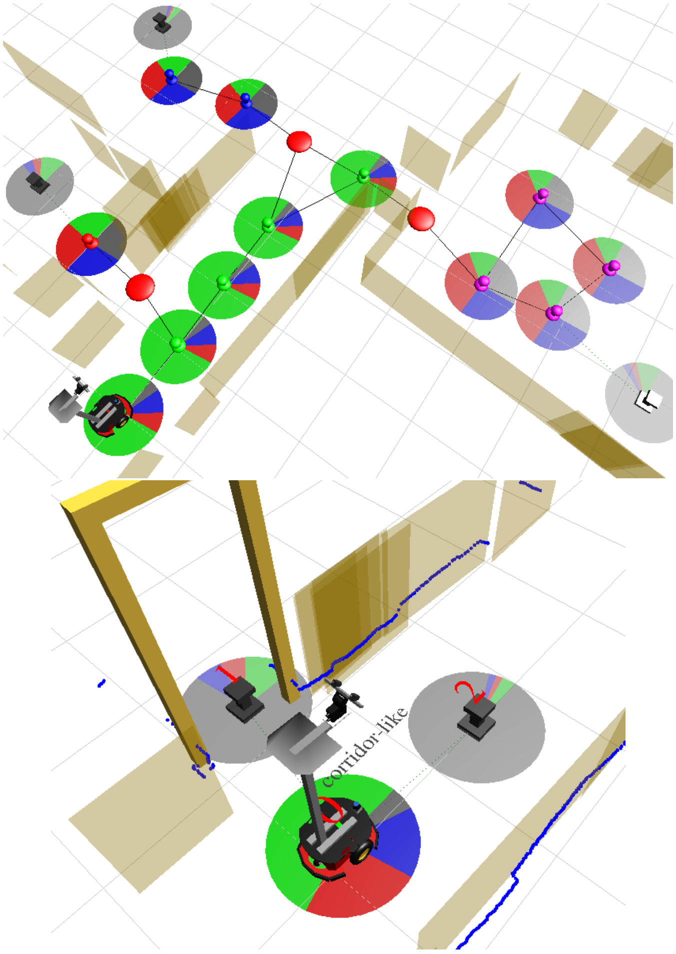


Figure 2.2: (a) A place map with several places and placeholders shown as large circular discs and 3 detected doors shown as smaller discs. The places have circular pins at the center of discs and placeholders have rectangular pins. Colors on discs indicate the probability of a place being in a room of a certain category in the form of a pie chart. Here green is *corridor*, red is *kitchen* and blue is *office*. (b) The start of a search run where two placeholders are detected with different probabilities of leading into new rooms of certain categories. The size of the color indicates the probability that the placeholder leads to a new room of a certain category (grey represent the case that there is no new room). One of them is behind a door hypothesis therefore having a higher probability of leading into a new room.

- $p(C^{c_i}_{h_j}|\boldsymbol{\theta})$, $C^{c_i}_{h_j}\in\{0,1\}$ - distribution indicating whether the placeholder h_j leads to a new room of the category c_i upon exploration. The knowledge about unexplored space is derived from all the observations $\boldsymbol{\theta}$ gathered by the robot in the explored part of space.

This information can be used to decide whether to explore one placeholder instead of another or simply perform fine-grained search for an object in one of the previously discovered rooms. A visualization of the distributions is presented in Figure 2.2.

Assigning Probabilities

In order to calculate the aforementioned probability distributions for the partially explored environment we used the probabilistic semantic mapping framework recently proposed in [4]. Though the specific semantic mapping framework is not a contribution of this chapter, we will explain it briefly so that the presentation in this chapter is self-contained.

The joint distribution representing the dependencies between object categories and room categories for known rooms is modeled as a probabilistic chain graph model [62]. This is due to the complex relationships between the entities in semantic maps. As an example, while we can assume to have a causal relationship between the shape of a room and it's category (if a room is determined to be elongated from sensory data, then it's likely to be a corridor), room-room connections in the graph may not have such one way causality (rooms connected in topology effects each other's categories both ways). Chain graphs are deemed as a suitable representation to describe these relationships, as well as the underlying topological nature of the world in one probabilistic representation. The structure of the graph model is presented in Figure 2.3 and is adapted at run-time according to the state of the underlying topological map.

The semantic mapping framework relies on several properties or attributes of space obtained from various modalities. Those properties characterize each of the places and contribute to the knowledge about room categories. We use the following properties in our implementation: geometrical room shape and size obtained from laser range data and general room appearance captured by a camera. In the chain graph model shown in Figure 2.3, the values of those properties are represented as a set of variables $(SH_{p_i}, SI_{p_i}, A_{p_i})$ for shape, size and appearance properties respectively. These properties are generated for each newly discovered place as the robot moves through the environment.

The spatial property variables for all places in a single room r_j are connected to a random variable C_{r_j} representing the functional category of the room. The relations between place properties and room categories ($p_{sh}(SH_{p_i}|C_{r_j})$, $p_{si}(SI_{p_i}|C_{r_j})$, $p_a(A_{p_i}|C_{r_j})$) are derived from the default knowledge. The shape, size and appearance properties can be observed by the robot in the form of features extracted directly from the robot's sensory input. The links between observations and the

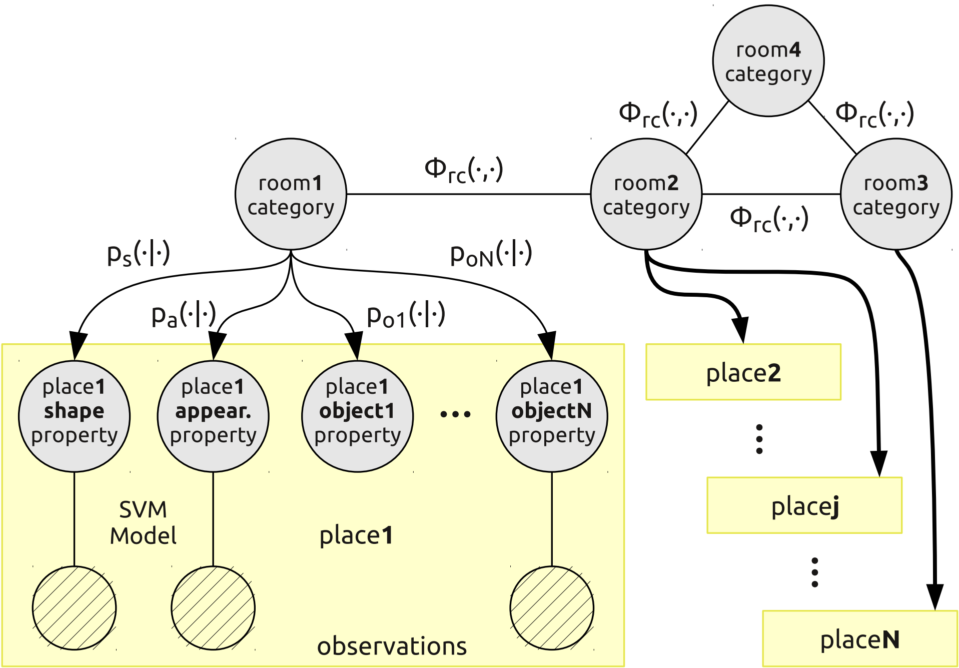


Figure 2.3: Structure of the chain graph model representing the search space at the large scale. The vertices represent random variables. The edges represent the directed and undirected probabilistic relationships between the random variables. The textured vertices indicate observations that correspond to sensed evidence.

place property variables are quantified by categorical models of sensory information implemented using Support Vector Machines [4].

Additionally, for each room, there is a set of variables representing the presence of a certain set of objects of each category in the already searched space inside the room $(O_{r_j}^{o_1}, \dots, O_{r_j}^{o_{N_o}},\ O_{r_j}^{o_i} \in \mathbb{N}_0)$ (e.g. for reasoning about finding another cup in a kitchen, after having found one cup.). Those variables are linked to the corresponding room category variable C_{r_j}. This relation represents the default knowledge about canonical object locations (e.g. that a coffee machine is likely to be found in a kitchen). The values of the object variables are directly observed and set to a certain value depending on the number of objects of a certain category detected in the room.

Finally, the potential functions $\phi_{rc}(C_{r_i}, C_{r_j})$ describe knowledge about typical connectivity of two rooms of certain categories (e.g. that kitchens are more likely to be connected to corridors than to other kitchens). Those connections propagate semantic knowledge between rooms represented in the topological map.

The default knowledge about room connectivity, shapes, sizes and appearances was acquired by analyzing annotated databases typically used for experiments with place categorization [4]. The databases consist of floor plans and images captured in various environments labeled with room categories as well as values of spatial properties (shape, size, general appearance). The conditional probability distributions $p_{o_i}(O^{o_i}_{r_j}|C_{r_j})$ relating the number of objects ($O^{o_i}_{r_j} \in \mathbb{N}_0$) of a certain object category o_i present in an already searched part of a room, r_j, to the category of r_j (C_{r_j}) are represented using Poisson distributions (e.g. the probability of finding another cup in a kitchen after having searched for one). The rationale behind this is that each occurrence of a certain object category in a room is conditionally independent from each other, with an expected total number of objects for that room category. The Poisson distribution allows us to model the expected number of object occurrences in a room through its parameter λ:

$$p_{o_i}(k|c_j) = \frac{(\beta\lambda_{o_i,c_j})^k e^{-\beta\lambda_{o_i,c_j}}}{k!}. \tag{2.1}$$

The parameter k is the actual number of object occurrences that we are interested in (e.g. what is the probability of finding *two* books in this room?) and β indicates the percentage of the room already searched by the robot (e.g. half of the room). In our model, λ_{o_i,c_j} is estimated separately for each object type and functional room category. It is calculated from the probability of existence of an object of the type o_i in a room of category c_j obtained from common-sense knowledge databases. The process is first bootstrapped using a part of the *Open Mind Indoor Common Sense* database[4] from which potential pairs of objects and their locations are extracted. Those pairs are then used to to generate '*obj* in the *loc*' queries to an online image search engine. The number of returned hits is then used to obtain the probability value. More details about this approach can be found in [63]. Once the probability of existence of an object of a specific type in a room of a specific category is obtained, the λ_{o_i,c_j} is calculated so that $\sum_{k=1}^{\infty} p_{o_i}(k|c_j)$ is equal to that probability.

Given observations of some of the objects and properties of space for the explored part of the environment, the distribution $p(C_{r_j}|\boldsymbol{\theta})$ over room categories of a room r_j can simply be calculated by marginalizing over all other variables in the chain graph model. Below we describe the models used for reasoning about unsearched and unexplored parts of the environment.

Reasoning about Unsearched Parts of the Environment

Given the model built for the explored and searched part of the environment, we can now use it to reason about the presence of objects and rooms in yet unsearched or unexplored space behind a placeholder. To this end, the chain graph model is extended in two ways.

First for the unsearched space, as shown in Figure 2.4, we add a set of variables $O^{o_1}_{s,r_j}, \ldots, O^{o_{N_o}}_{s,r_j}$, $O^{o_i}_{s,r_j} \in \{0,1\}$ which allow us to reason about the presence of

[4] http://openmind.hri-us.com/

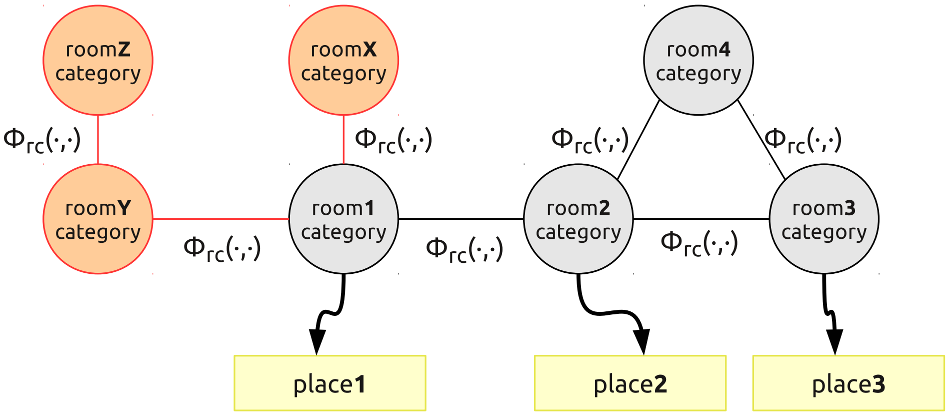


Figure 2.4: Examples of extensions of the search space model permitting reasoning about unexplored space behind placeholder located in room 1.

objects of various types in unsearched parts of known rooms. The distributions $p_{s,o_i}(O^{o_i}_{s,r_j}|C_{r_j})$ are represented in a very similar fashion to Eq. 2.1, however this time focusing on the remaining, unsearched portion of space $1-\beta$. Since, in order to direct the search, we are only interested in the presence of at least one instance of the object, $p_{s,o_i}(O^{o_i}_{s,r_j}|C_{r_j})$ simplifies to:

$$p_{s,o_i}(O^{o_i}_{s,r_j}{=}1|C_{r_j}{=}c_l) = 1 - e^{-(1-\beta)\lambda_{o_i,c_l}}. \tag{2.2}$$

Then, the probability $p(O^{o_i}_{s,r_j}|\boldsymbol{\theta})$ is obtained by marginalizing over all the other variables in the chain graph model.

Second, in order to reason about unexplored space behind a placeholder, we hypothesize potential room configurations in the topological map of the environment. For each configuration, we extend the chain graph from the room in which the placeholder exists with variables representing categories of hypothesized rooms. Then, the categories of the hypothesized rooms are calculated by performing inference on the chain graph and the probability of existence of a new room of a certain category behind the placeholder is obtained by summing over the room category inference results for all possible configurations.

In our system we consider three hypotheses[5]: (1) placeholder leads to a single new room; (2) placeholder leads to a new room connected to another new room; (3) placeholder does not lead to a new room. For the cases (1) and (2) we extend the chain graph model, as shown in Figure 2.4, by adding additional room category variables C_{r_X}, C_{r_Y} and C_{r_Z} connected to the variable representing category of the room in which the placeholder is located. The probability of there being a new room

[5] These are based on the observation that in typical indoor environments you can reach most rooms in two steps thanks to ”connector rooms” like corridors

of a certain category c_i behind the placeholder h_j is then calculated as follows:

$$\begin{aligned} p(C^{c_i}_{h_j}{=}1|\boldsymbol{\theta}) = p(r^1_{h_j})\; p(C_{r_X}{=}c_i|\boldsymbol{\theta}) + \\ + p(r^2_{h_j}) \sum_{y=i \vee z=i} p(C_{r_Y}{=}c_y, C_{r_Z}{=}c_z|\boldsymbol{\theta}), \end{aligned} \tag{2.3}$$

where $p(r^1_{h_j})$ and $p(r^2_{h_j})$ are priors assigned to each of the hypotheses. If we assign equal prior to the case (1) and (2), it is sufficient to calculate a probability of the placeholder leading to at least one room ($p(r_{h_j})$). This can be estimated as follows: $p(r_{h_j}) = p(h_{h_j})(1 - p(d_{h_j})) + p(d_{h_j})$, where $p(h_{h_j})$ denotes the probability that the placeholder h_j leads to another placeholder and thus potentially to another room and $p(d_{h_j})$ is the probability of there being a doorway behind the placeholder obtained from a door detector. The value $p(h_{h_j})$ can be estimated from the amount of open space in the direction of the placeholder estimated from the laser range data. The outcome can be seen in Fig. 2.5.

Modeling the Search Space on the Room Scale

We maintain a 3D metric map for each room which supports viewpoint selection for object search as well as obstacle avoidance and path planning. This map is represented as a 3D grid consisting of equally sized grid cells. Each cell holds the occupancy information and the probability of the target object being in this cell as in [26].

The sum of the probability value of all the cells given a room comes from the chain graph model, namely the estimated value of $p(O^{o_i}_{r_j}|\boldsymbol{\theta})$ described earlier. The total probability is uniformly distributed over all occupied cells as possible locations for the target object's center point, see Fig 2.5. This way we connect the object probabilities at the place map to the finer 3D metric representation of the same space.

Furthermore, changes in this 3D spatial probability distribution should also influence the probabilities in the place map. As an example, processing a viewpoint inside a room without finding an object reduces the probability values of the cells that are visible from this viewpoint. This change needs to be reflected in the place map as well since the decision making algorithms need to operate on the place map level and not at the fine grained 3D map level, for reasons discussed in Section 2.4.

To this end, as explained previously, we introduce the term β to the chain graph model which represents the ratio of the space searched by the robot in a room. By updating this value accordingly during the search process, the system can reason about the trade-off between continuing to search the current room or execute another action such as exploration or search in another location.

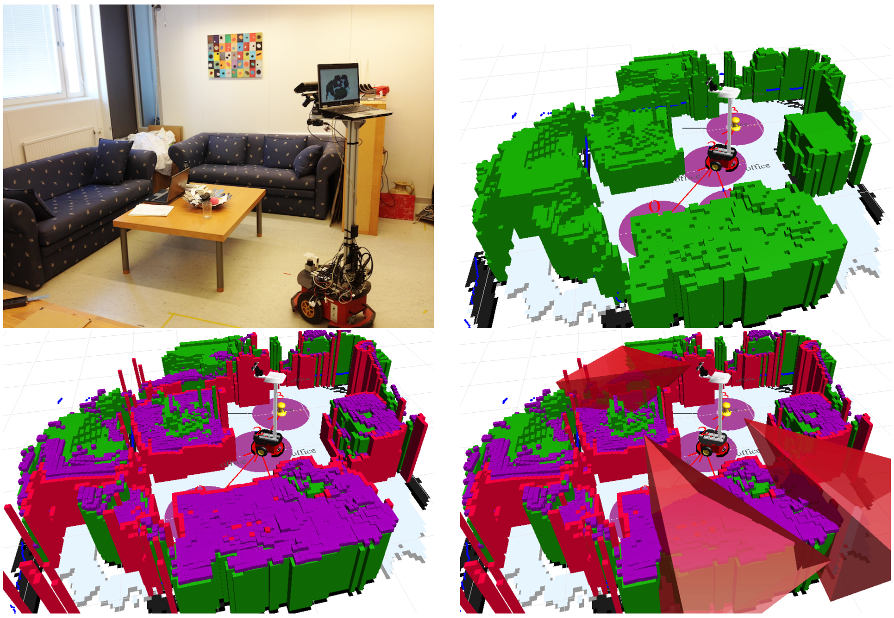


Figure 2.5: (a) The robot during a search run in a room (b) Corresponding 3D map of the room built by the robot, where green areas represent obstacles. (c) The spatial probability distribution over 3D space shown in purple, where occupied regions of space has a likelihood of containing the object. (d) Three viewpoints computed for this room is shown.

2.6 Experiments

Experiments took place in a 33m×12m environment with 18 different rooms of which 16 are offices, 1 is kitchen and the remaining room is a meeting room connected by a corridor. The mobile robot platform utilized is a Pioneer III wheeled robot, equipped with a Hokuyo URG laser scanner, a Microsoft Kinect camera and a higher resolution camera mounted at 1.4 m above the floor.

The robot had models of all objects it searches for before each search run. Three different objects are used during experiments; cerealbox, stapler and coffee mug (Fig. 2.7). The default knowledge indicates that the cerealbox is mostly expected to be in a kitchen, the stapler in office rooms and the coffee mug can be almost in any room in the environment except corridor.

Experimental setup

It is generally very hard to obtain ground truth data on the task of searching for objects in large environments for quantitative analysis of the system. There are sev-

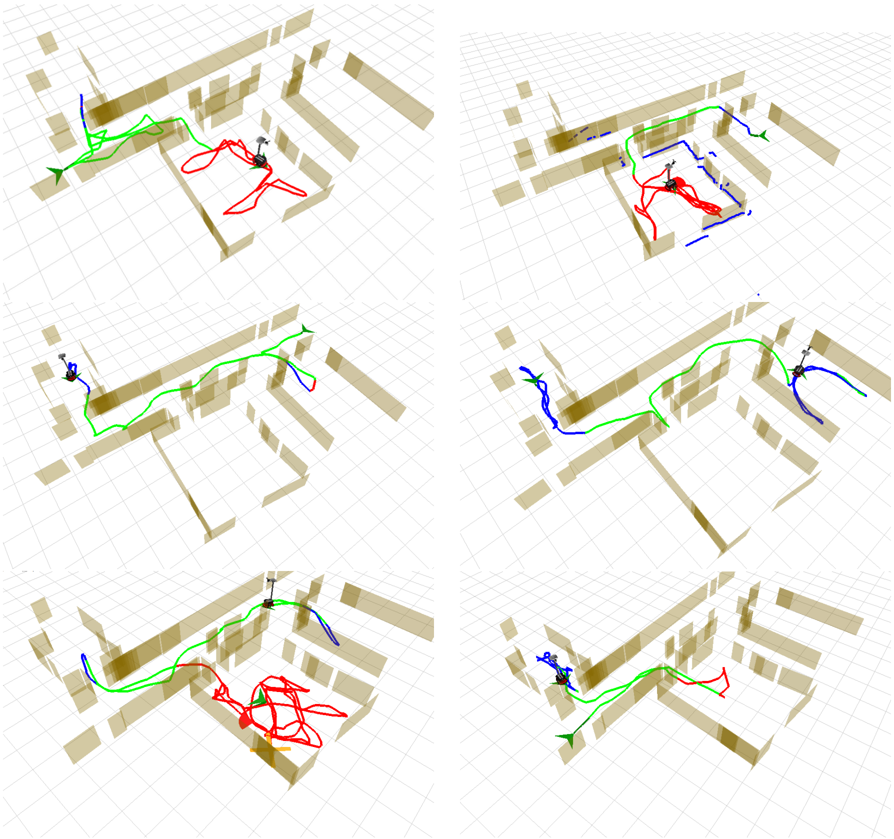


Figure 2.6: Search trajectories resulted with the method presented in this work in an environment with three rooms.

eral reasons for this. First, there are no established datasets as the active nature of the problem makes this hard and there are no well established simulation environments in the literature in order to compare the few systems that are designed to search for objects in large scale environments. Second, in the absence of a benchmark dataset for visual search tasks, different systems need be evaluated under the same conditions (e.g. same environment, same objects and object placements) for a meaningful comparison of results.

We have implemented a method for comparison, later referred to as *uninformed search*, which doesn't make use of the semantics of the environment. With uninformed search, we aim to recreate the greedy search strategy employed in the most recent the state-of-the-art systems on active visual search such as [28], [27] and [29]. In this case the robot, at each newly discovered room, first explores the room's extent and calculates viewpoints that covers the entire room. The robot then proceeds to process each viewpoint one by one, in a greedy fashion. The search

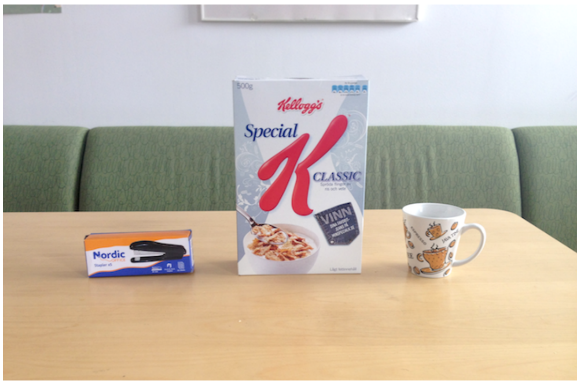


Figure 2.7: Objects used in the experiments.

continues until the object is found or there are no room left that aren't searched.

Furthermore, we also compare our method to a human performing the active visual search task. We believe this provides a gold standard on testing the efficiency. The idea of comparing an object search method against human participants has been explored recently in [47]. Our work differs in that we let the human remote control the robot and perceive the environment using the same sensors as the robot.

Finally, we have implemented the method proposed in this work, later referred to as *informed search*, that uses the semantics of the environment to guide the search task by using the actions and the spatial representation presented in this work.

Our experiment setup is as follows. We invited 12 people to the Center for Autonomous Systems (CAS) Laboratory. The participants are picked such that they had not seen the test environment beforehand and were not familiar with this work or robotics in general.

First, the target objects were shown to each participant from all view angles. This corresponds to learning the object models in the system. Then, each participant is given a driving practice of steering the robot with a joystick at another location, until they are comfortable in maneuvering the robot. The participants passed certain tests which proved that they were able to drive the robot comfortably. This included moving between rooms and to designated places in the environment. Once the participants were able to control the robot with ease using the joystick, they then sat in front of a computer which displayed a live feed video from robot's cameras. The robot is placed in the test environment which is the entire 6^{th} floor of CAS. The participants were then asked to find one of the objects with the starting point for the robot and the object location is picked randomly for each run[6].

[6] The cerealbox object was placed only in the kitchen, the stapler was placed in one of the office or meeting rooms and coffee mug was placed in any of the rooms except corridor, commensurate

After a participant has completed a run, we changed the location of the object and asked the participant to perform another search task. This corresponds to the case when the robot has a known map of the environment at the start of the search task. For each of the human runs we have run informed and uninformed object search methods in exactly the same conditions regarding robot's starting position, presence or absence of an a priori map and location of the target object. Repeating this process for the three objects, in total this resulted in 108 real-world test runs of an object search task for all three methods.

Quantitative experiments

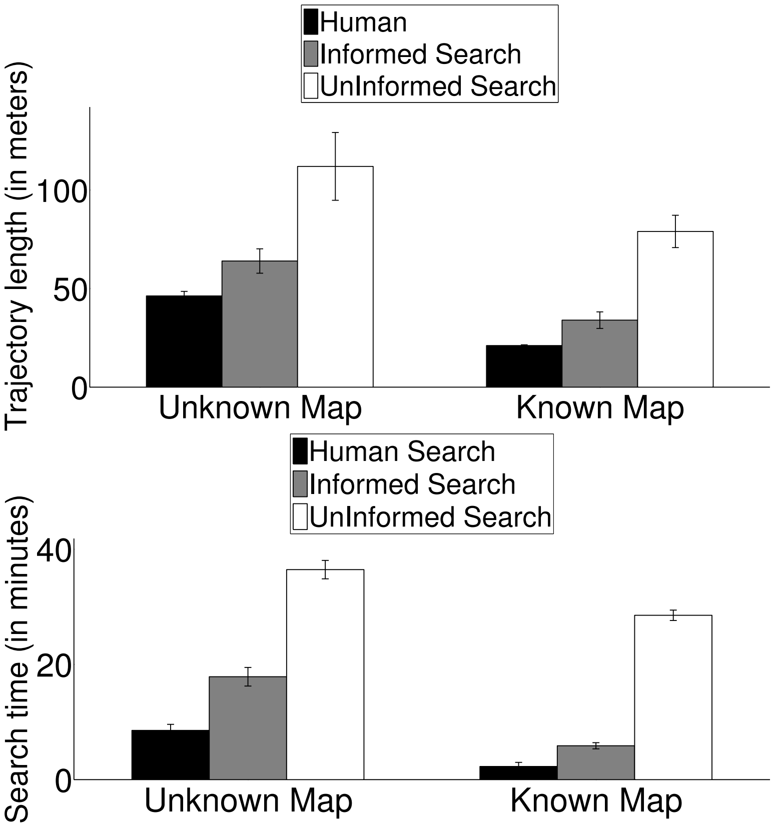
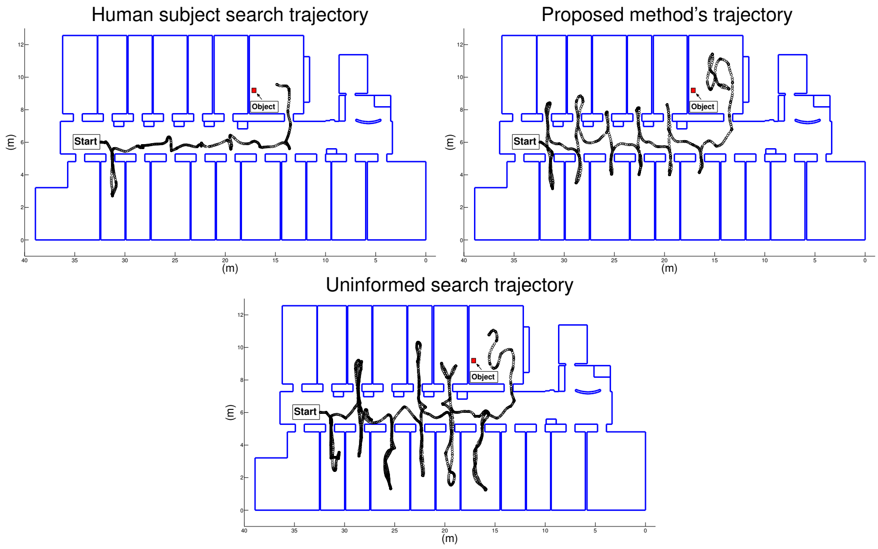


Figure 2.8: (a) The average trajectory length and (b) average search time for human, uninformed and informed search methods for both the unknown and known map cases over 108 search runs.

with the default knowledge.



Human subject search trajectory

Proposed method's trajectory

Uninformed search trajectory

Figure 2.9: Trajectories overlaid on the floorplan of the test environment from an visual search run for the object cerealbox with unknown map. 2.6 is the trajectory for one of the human runs, 2.6 is the trajectory taken by the method presented in this work and 2.6 shows the trajectory of the uninformed search method.

We have recorded the search trajectory and time during runs for both the unknown map and known map cases. Figure 2.8 shows the average trajectory lengths and search time for all three cases for both when the map is known a priori and unknown. To compare with the robot performance, we have only considered instances where the object is successfully found by the robot. The uninformed search is by far the most inefficient way of locating objects by traveling on average 112 meters. In contrast, with the use of semantics of the environment in order to guide the object search, the robot nearly halves the total search trajectory length. As expected, human runs have the shortest trajectory length. It is worth noting that the difference between the human performance and the informed search is smaller when the map is known. This shows that the method presented in this work can utilize the already known information about space without additional algorithmic or implementation changes.

The difference in trajectory length between the human and and the informed search methods is approximately 25 meters in the unknown map case and 18 meters in the known map case. The search strategy of human participants is the very similar to the method presented in this chapter but humans are much better in visual processing. Therefore we would expect that the search strategy and modeling of the search space presented in this work would automatically benefit from better visual processes in categorizing rooms and detecting objects in images.

Looking at the object search task performance from the search time point of view, we see a similar ordering in terms of the three methods tested. The differences between the methods are larger than in the trajectory length case. The reason for this is, moving more in the environment requires turns and twists which takes a longer time that doesn't manifest itself in the trajectory length metric. As can be seen from the graph, in the uninformed search case, it approximately takes 36.4 minutes to find the target object for the unknown map case and 28.5 minutes for the known map case. This is clearly an unacceptable time to wait for an intelligent autonomous robot living in human spaces. On the other hand, by utilizing the semantics of the environment as described in this work, on average we are able to find the object on average 15.8 minutes in the unknown map case and 7.8 minutes in the known map case. This is a huge gain in efficiency. As expected, in the known map case, human participants were very adept at learning the environment. The search time for the human participants for the unknown and known map cases were 8.55 minutes and 3.6 minutes respectively. We note that some of the difference in time between the human and informed search is caused by humans navigating the robot more expertly around obstacles than the autonomous navigation algorithm.

Figure 2.9 shows the trajectories of human, uninformed and informed search methods from a single run. The target object is the cerealbox and the map is unknown. The starting position for all of the runs is the same, the leftmost position in the corridor indicated in Figure 2.9.

The search trajectory of the human participant is shown in Figure 2.6. At the start of the search, the human participant steered the robot into the office room. This was a commonly occurring behavior with most human participants. We argue that the reason for this is that the participant first required some initial information on the type of the environment. Upon realizing that the room is a typical office room, and most likely all other office rooms in this floor look alike, the participants typically have constructed their idea of what kind of an environment this is, therefore what types of rooms they should anticipate. Therefore, for the remaining part of the trajectory they only peeked through the doors of other rooms to inquire its category, until they have found the kitchen room. The cerealbox object was placed on a table in plain sight in the kitchen.

Figure 2.6 shows the search trajectory of the method presented in this work. The robot here starts out in the corridor and by exploration it enters each room. One difference between the human case is that, while humans can deduce a room's category by peeking through the doors, the robot has to enter the room and accumulate observations. However, after gathering evidence that the category of the room is not what the planner expected, the robot continues with the exploration in the corridor with the hopes of finding a kitchen room. As expected, the informed search method traverses significantly less amount of trajectory compared to the uninformed search method.

Finally, Figure 2.6 shows the uninformed search trajectory. As expected this strategy covers the whole environment without making use of spatial characteristics of the visited places. This results in a significantly less efficient search compared to other two methods.

Qualitative experiments

In order to show the adaptability of our system to different search conditions and provide better understanding of typical search missions, we have also run our system in a smaller, 3-room environment and logged the input/outputs and the resulting trajectories. We have evaluated six different scenarios with different starting positions, object locations and map status (from completely unknown to partially known at the start of search) each time. Figure 2.6 shows the search trajectory of all the runs. The trajectories are color coded. The colors indicate robot's room category estimates for the current position; red, green, blue corresponds to kitchen, corridor and office respectively. In the following we give a brief explanation for what happened in the different runs:

- Fig. 2.6 Starts: *corridor*, Target: *cerealbox* in *kitchen*
  The robot starts by exploring the first placeholder in the *corridor*. After this, two more placeholders appear, one continuing in the corridor, the other on the left behind a doorway. The latter placeholder has a higher probability of leading into a kitchen due to it being nearer to the doorway and the robot enters *office1*. The robot then starts exploring other placeholders appearing in *office1*. After exploring the second placeholder in *office1*, the room category of *office1* is deemed as *office*. Since default knowledge indicates cerealboxes are seldom in office rooms, the robot returns to exploring the corridor until it finds the *kitchen* door and explores the placeholder near this door. This time, exploration of kitchen goes without interruption since cerealboxes have a high probability of being located in rooms with category kitchen. Finally, the robot computes views in this room with the CalculateViews action. After processing some view positions, the cerealbox object is found.

- Fig. 2.6 Starts: *office2*, Target: *cerealbox* in *kitchen*
  Similar to Fig. 2.6, after exploring a few place holders, the robot does not issue the search command in the current room, and continues with exploration until it finds the corridor. Eventually, the robot finds the room kitchen and the rest proceeds as in Fig. 2.6.

- Fig. 2.6 Starts: *corridor* Target: *cerealbox* in *kitchen*
  The robot explores until it finds *office2*. Upon entry the robot categorizes *office2* as kitchen but after further exploration, *office2* is categorized correctly. As a result of this, the robot switches back to exploration and since the kitchen door is closed, it passes the kitchen and finds *office1*. Similarly after determining the category of *office1* the robot sets out to explore more however there are no more placeholders to explore therefore the search is stopped.

- Fig. 2.6 Starts: *office1* Target:*stapler* in *office2*
  The robot starts by exploring the current room and meanwhile categorizes

the room correct as an office room. Since *stapler* has a high probability of being in offices, the robot launches a search in this room. However the object is placed in *office2* and the robot fails to find the object. After failing to find the object in *office1* the robot continues with exploration, which it leads to the corridor. The robot then finds the room *kitchen* but after realizing that it is kitchen-like decides not to search the kitchen room and continue with exploration still. The robot then finds the room *office2*. After determining the category of this room, the robot launches a search and this time the *stapler* object is found in *office2*.

- Fig. 2.6 Starts: *kitchen* Target: *cerealbox* in *kitchen*
  As before, realizing that the current room is promising for cerealbox, the robot calculates viewpoints in this room. After processing the views its visual algorithms fails to detect the object. After processing all views, it finally goes out in the corridor to look for another kitchen. However the environment is fully explored and the search stops. This is a case where the search strategy has successfully brought the object in the field of the view of the robot however there was a failure in object detection.

- Fig. 2.6 Starts: *corridor* Target: *stapler* in *office1*
  The robot is started in the corridor and driven to the kitchen by a joystick; thus in this case the environment is largely explored already when the planner is activated. The part of the corridor leading to *office2* has been blocked deliberately. By exploration, the robot finds its way to *office1* and launches a search which results in a successful detection of the target object.

Comparison to previous work

In this section we will compare our approach to those in previous work that are closest to our work. In short, we will focus on three different lines of work [26, 29], [32] and [31]. We note that a quantitative comparison is not possible since it is either not possible to recreate the exact search environments that these works have produced their results, obtain the same target objects or search conditions. Therefore we will aim to give an extensive discussion on how the method described in this work presents a contribution in the light of these works.

The pioneering work by Tsotsos et al. in active visual search with mobile robots introduced the first ideas on view planning in 3D space with a moving agent and a spatial probability distribution defined over the search space [26]. In later work from the same authors, the environment is assumed to be unknown in advance as it is in this work [27]. The robot exhaustively covers the search space in this work until the object is found or the whole environment is covered. A very recent visual search system presented [29] uses a similar greedy search strategy. The uninformed search method implemented and evaluated in this work approximates to this type of search. As shown, such a method is highly inefficient for the search space and target objects depicted in this work. In contrast, we utilize semantics of the environment to prune the search space and guide the search towards the more promising areas of the search space.

The system described in [32] takes on a different approach by first identifying candidate positions in the search environment that have visual features similar to that of the target object. After this first step, the robot then revisits each of these positions to run a more computationally expensive and powerful object recognition algorithm. While this approach exploits the visual similarity, it still first needs to cover the whole space in order to generate candidate locations. In our case, this would mean exploring the entire floor which would be prohibitively costly from a task completion point of view.

Finally the work by [31] is closest to our work in the sense that environment semantics are utilized to guide the search. In this case, object-object co-occurrence properties are exploited in order to compute paths in the environment that leads to the target object. In order to accomplish this, the environment is first explored and various objects are discovered. These objects later indicate where the target object might be in the environment. As an example if the robot is looking for a chair then an area where there are lots of tables can be a good candidate place. A path is computed to this area. There is no view planning involved, the target object is deemed found if the images while traversing the path contains the said object. In comparison, our method doesn't rely on first exploring and discovering a dense set of objects in the environment. Instead we utilize default knowledge about object locations and room categories (e.g. cerealbox is likely to be found in the kitchen) to guide the search.

2.7 Conclusion on the proposed AVS system

In this chapter, we have used the AVS application as a vehicle to investigate how spatial structure can aid the performance of challenging tasks in robotics. Active visual search in large environments remains an unsolved problem with recently increasing interest. By exploiting the inherent structure in the environment, we have proposed a search strategy that allows for efficient active visual search in large unknown environments that can handle all of the following points for the first time in the literature:

- Large search spaces at the scale of building floors
- Unknown or partially known map
- Visual search with a limited field-of-view camera in a 3D real-world environment

We have argued that by exploiting the semantics of the environment, a search strategy that effectively reduces the search space, allowing the otherwise intractable search problem to be solved, can be devised. To accomplish this, we have demonstrated how to build and use uncertain environment semantics in order to efficiently search for objects. Further, we have proposed a way of dealing with unexplored environments by reasoning about possible worlds in the same spatial model. We define search actions that allow efficient search over the whole environment by taking full advantage of this model of the search space. We show that this set of actions allow a flexible search system that can handle different starting positions and environments with ease. We have compared the search method proposed in this work to two other methods. First, we have implemented the greedy coverage-based search method, similar to current state-of-the-art methods, where no semantic information is utilized. Second, we have carefully designed human search experiments, which would arguably be the most efficient and successful object search system in the world currently. We compared the trajectory length and time. Our results show that the proposed method significantly outperforms greedy search and using semantics of the environment, an object search task can be done much more efficiently.

2.8 Lessons learned: Are we making use of spatial structure enough?

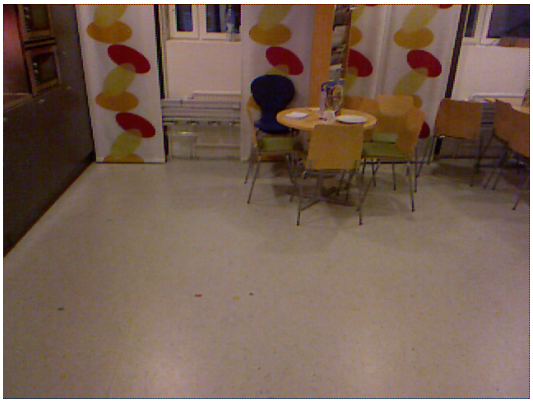


Figure 2.10: An image frame captured while a mobile robot enters a kitchen.

The goal of the work behind this chapter was to pick a challenging task in order to gain more insight about the questions laid out in Chapter 1. The proposed improvements on the AVS problem by using the semantics of man-made environments encourages us to determine more areas where exploiting structure can help. As a result of devising and building such a system we have noticed two ideas where spatial structure is underutilized spanning both the small scale (e.g. in a single image or a scene) and large scale (e.g. an entire building floor).

First, imagine the case in figure 2.10, where a mobile robot looking for a coffee mug enters a kitchen. In this scene, we can clearly see a table and focusing on the table we can spot the white coffee mug, however currently state of the art AVS methods such as the system presented in this chapter are far from this level of performance. In this case, since the white mug is hard to spot, depending on the parameters of the recognition algorithm, one would either get a detection on the object but at the cost of a high number of false positives or no detection at all. On the other end of the scale, if the parameters are tuned to have a higher true positive rate instead, we would miss the object completely. However, looking at the scene, even though the object itself is hard to spot, we can still clearly see the table, the floor and the walls as well as the chairs. From common-sense, we can know that coffee mugs don't just appear to be glued to walls and back of chairs and they seldom appear to be on the floor or near the ceiling height. Armed with all these priors, looking at the image, we *know* where the coffee mug should be at, if there's one indeed in this image. Then, we can either focus in those parts of the scene or move ourselves to have a closer look at the table top. These are all useful priors based on the inherent structure of indoors that we have become accustomed with.

In an embodied robotic system [7] equipped with various sensors we would expect such prior information to help in detection of objects. We think this is a case where structure about the world can greatly help in improving the understanding of these type of scenes.

Second, imagine the case where we enter a new building looking for a meeting room. Although we were not in this particular environment before, we would know that corridors connect rooms to each other, there's probably only a few meeting rooms and not more than there are offices, and that they are typically in the middle of a floor and not in a remote corner of the building. Most often we assume that these assumptions are safe to make, as the structure of indoors at the floor level indicates they mostly hold true. In the rare occasion that they aren't, we can always fall back on exhaustive strategies. Not only being true, but such priors help us getting to the meeting room much faster than randomly walking around.

Similarly, coming to a robotics setting, imagine a robot tasked with finding an object. The robot starts in a single room as was the case in AVS application presented in this chapter, aiming to find the kitchen to search for an object. At this point we should ask ourselves: How does the robot know that rooms are typically connected with corridors to each other, and there's typically one kitchen on a floor which can be reached by following the corridor, but not ten? How can we equip our robots with a deeper understanding of indoor environments?

Currently, to the best of our knowledge, no system exploits such crucial information about how indoor environments are configured. However we posit that there's inherent structure in how indoor environments are configured at large scales, i.e. how rooms and building floors are laid out. We think that discovering and ex-

[7] What we want to emphasize here is that, a robot is a physical entity and not a bodiless *sensor in the sky*. We can in advance know what is up and down, the height of which a camera is mounted at and its pose in the relative frame of the robot among other things. Therefore, we should use all the benefits of being an embodied system.

ploiting structure in man-made environments at the large scale is very beneficial to autonomous robots and has escaped the attention of robotics thus far.

In the next chapters, we will thoroughly analyze methods for discovering, modeling and exploiting structure in man-made environments across multiple scales and modalities.

Chapter 3

Modeling and exploiting local 3D geometry for predicting object locations in indoor scenes

As we have established in the previous chapter, locating everyday objects in indoor environments is the prerequisite of many robotics tasks such as mobile manipulation and semantic mapping. Finding objects in large environments is a challenging task considering that at every point in time during the search, while the robot is moving, images acquired by the camera can be useful to the task and ideally should influence the search behavior.

Imagine the case where you pass by a kitchen area and notice the coffee mug with the corner of your eye and turn your gaze to locate and grab the said mug. Ideally, we would want an object searching robot to also exhibit such reactive and smooth search behavior (as opposed to computing few sparse view points in advance and only analyzing images acquired from these view points). However there exists multiple hurdles in processing the images acquired in a continuously moving robot due to the nature of quick movements in an everyday setting. Amongst the most problematic, the object typically occupies very little space in the images taken by a mobile robot in motion (especially in the sequence images where the target object first enters the robot's field of view), different camera angles and cluttered scenes make detection unreliable and there's significant motion blur or severe occlusions prevent a clear view of the object.

In this chapter, we want to explore one idea following what we have started with in Chapter 1, namely that autonomous systems that operates in man-made environments should take advantage of the structure of the world. In Chapter 2 we have realized an AVS system and discovered improvement areas. In this chapter, we investigate one of such idea stemming from Chapter 2.

While working on the active visual search system presented in Chapter 2, we have seen that typically state-of-the-art object detection algorithms that are designed for finding objects in images of a certain data set where the object of interest is either in clear view, close to the camera or not severely occluded have a hard time dealing with the stream of images acquired by a moving mobile robot. This is in line with what we see when we look at highly popular datasets in object detection and recognition across different object classes.

Further and more importantly, the methods designed for such test images often assume that we're looking at the world from a *bodiless camera* and not from the point of view of an embodied agent, which has a specific function and purpose, which operates in specific environments which most likely has some kind of order (e.g. objects in man-made environments are not scattered around randomly.).

In contrast, we believe, in robotics scenarios, we should exploit the fact that the camera is mounted on an embodied robot, that is likely to be exposed to only a limited number of circumstances (e.g. an office robot will not need to deal with scenes that includes beach scenes) tightly embedded in its environment [64].

The main claim we make here is that the placement of everyday objects in indoor scenes is highly correlated to the local 3D geometry around these objects. We think that by utilizing such inherent structure, we can direct the robot's visual processes (such as object detection and recognition) to scene regions where there is high likelihood of there being an object of a specific type. This is in contrast with trying to find the object *anywhere* in the scene disregarding cues stemming from how human environments are typically configured.

In this chapter, our aim is to conceptualize and realize the idea of modeling and using local 3D geometry in object recognition tasks as one instance of utilizing the inherent structure in man-made environments.

As an example of the intuition behind the idea investigated in this chapter, let us look at Figure 3.1, which shows an image acquired by a mobile robot where the object cup first enters robot's field of view, as the robot enters the kitchen. The target object in this image is undetectably small. As we've explained, in such images, most object recognition algorithms which uses visual appearance features would either fail to detect the target object, or the detection would be accompanied by false positive detections.

However, appearance is likely not the reason for location of objects, i.e. cups are not typically found on tables because the former is red and the latter is brown, but the table offers physical support and easy reach. Generally, objects are placed in places where it is easy to interact with them; trash cans are typically on the floor and not on a high shelf. Furthermore, objects are placed to be physically stable at rest. As an example a cup almost never occur on a wall as there is no support for it there. We note that all these points to 3D structure of the world. We refer to the association between this geometrical structure and location as *3D context*.

Therefore, we claim that systems striving to efficiently locate an object should exploit *both* the shape of the object *and* the structure of the environment that these objects are part of. One obvious benefit of this in the context of localizing objects is that although the object itself may be small or not even visible, the supporting 3D shape might be bigger and detectable at a larger distance.

For realizing this idea, we leverage on the cheaply-available good quality 3D data brought by the recent advent of RGB-D sensors and show that the 3D context of objects is a strong indicator of object placement in everyday scenes. Figure 3.1 gives an example output, where the system has picked out a small region in the image around the cup, corresponding mainly to the table as the most likely region for a cup. No object recognition system would be able to recognize the cup itself at this distance in this image resolution. Additionally, plane fitting methods as employed in [65] might fail to detect the table plane as it occupies a very small part of the image. However, a system that exploits the local 3D structure in which objects typically occur can use this to reliably identify promising regions in the image for object presence.

3.1 Related work

There exists a large body of previous work investigating the use of appearance-based contextual information to improve object detection. Coming from a biological systems perspective, Bar has shown that visual objects of specific types often appear in some typical context [66] of their own. Bar argues that humans take advantage of contextual cues in order to facilitate recognition of visual objects by utilizing statistics on co-occurring objects. Further, human subjects often have a harder time recognizing objects placed in wrong contexts compared to objects appearing in images with no background or with the correct context. The author proposes a verifiable model for explaining how context is used by the brain. Bar's findings are in line with previous work on analyzing contextual cues' place on the time spent during visual recognition tasks in humans [67]. In this study the authors analyzed the eye movements of human participants to measure the effect of contextual priors and exactly at what stage they are utilized in the human object detection process.

The co-occurrence based contextual model requires parsing the initial scene into objects, which is a hard problem that requires high-level concepts such as objects and image segments. Instead Torralba et al. proposes a model where visual features extracted from the image as a whole (as opposed to local features that only describe a certain region of the image) [68]. These ideas are realized by the authors in a series of papers, demonstrating the usage of low-level global features for context driven attention and object detection [69, 70]. In their work, the general idea is to use appearance features in order to capture structural characteristics of scenes at a crude level such as spatial extent, perspective and openness among others. These only give a rough idea about the structure of the scene, whether or not the image depicts an open scene (e.g. a beach, plains) or a constrained, cluttered space (e.g. an office room, an urban scene). However, as the authors demonstrate, these cues work very well in predicting the location of a particular type of object in scenes, or pointing out to images where an object category might appear. As an example, if the image contains convergent lines, vertical structures on the sides and openness in the middle as in a road scene, pedestrians are likely to be on the sides of the

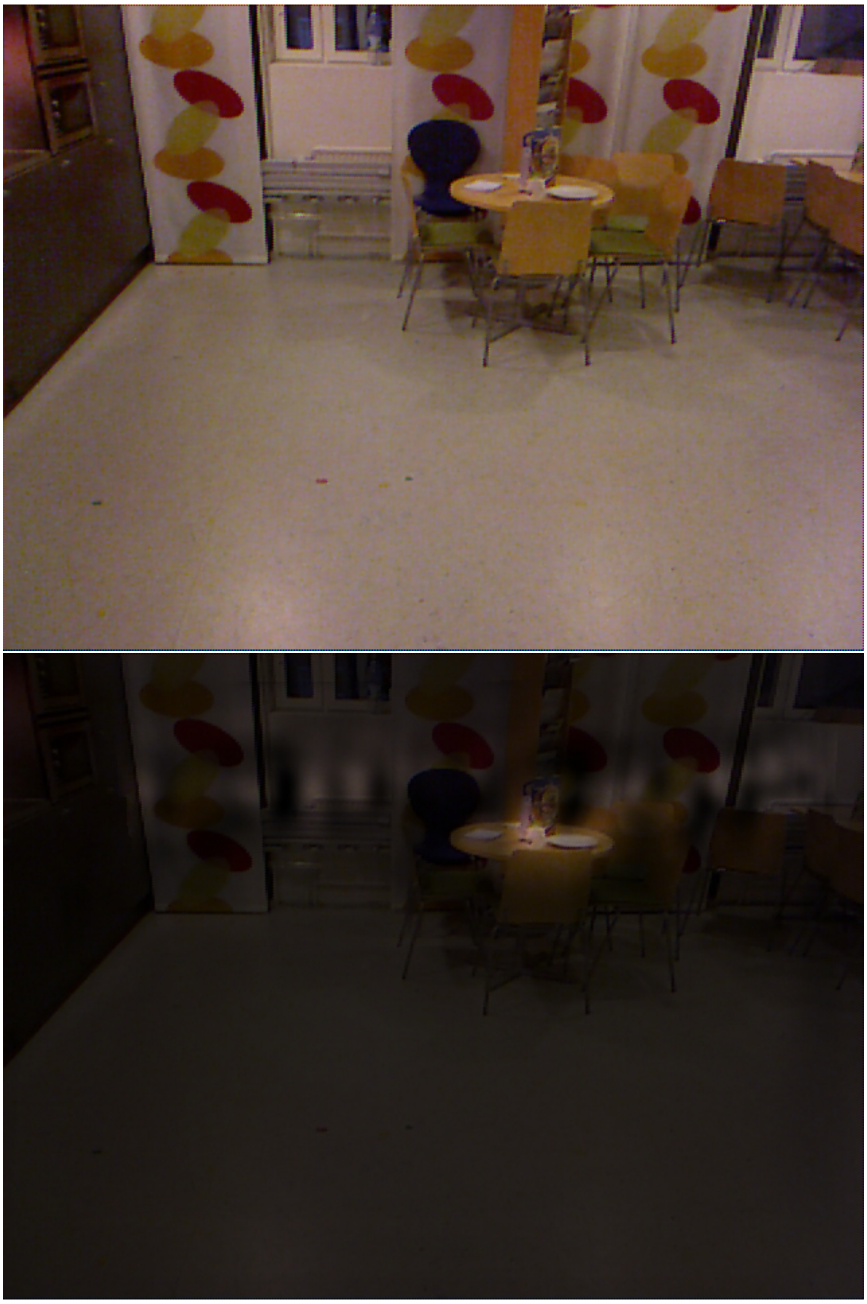


Figure 3.1: Best viewed in color. (a) A cup on a table within a larger room. The cup occupies a very small part of the sensory data. (b) The output from our method where bright areas correspond to high probability of object presence. Most object recognizers would scan the whole image and fail to find the object in this scene. However we can rule out large portions of the image by exploiting the 3D context in which objects appear.

images. Thus, the authors demonstrate that focusing search only on these areas can provide a boost in various performance metrics.

In [65, 71] the authors exploit the notion that the performance of object detection tasks can be boosted by extracting and tracking planar surfaces since a large number of everyday objects are located on tables and shelves. Even though this approach is robust in controlled environments, not all objects rest on flat planar surfaces of a certain size and developing tailor-made methods for each situation depending on the object class is not scalable.

In another line of research, models of visual attention mechanisms aim to locate salient parts of an image. The assumption is that the sought object stands out in the image, thus creating highly salient regions which attract visual attention where the visual processes are directed to [72, 73]. This mechanism can be used to prune the search space, where an object detector is only run on salient regions of the image. This has two main advantages. First, computationally expensive algorithms can focus a subset of the image, thus lowering the overall processing time. Second, false positives that may occur in non-salient parts of the images can be eliminated. The downside with this approach is that not all target objects are visually salient, as in the case of textureless or small objects, unless this is a requirement on the target objects [74].

Björkman et al. demonstrate an active vision system that attempts to find and fixate on target objects in scenes [75]. The authors have utilized 3D information about the scene extracted from stereo images and detect 3D blobs that roughly correspond with the known object size. These regions are then considered more likely to contain an object. In this manner, the system will only focus on parts of the scene that are likely to contain objects..

Similarly Frintrop et al. present a saliency-based object recognition method that works with depth images constructed from a tilting laser scanner [76]. In this work, salient regions from depth data and laser intensity are combined in order to provide initial location candidates for the target object.

In this work we draw inspiration from the idea of utilizing global image features as a way of describing the spatial extent of scenes [69]. As we noted previously, visual features were used as an indirect way of extracting the spatial layout of scenes. However, in this work, we note that we can directly utilize the depth data associated for each image from an RGB-D camera.

3.2 Exploiting Local 3D Context

Since we have established that context plays an important role in locating objects in scenes and introduced the idea of 3D context, in this section we describe in general terms how to make use of the 3D context to find likely object positions.

Most approaches to object detection look for the object features. Instead, 3D context is modeled from the surrounding of objects rather than the object itself. It therefore provides information of the type ”this is a likely place for the object” rather than "this is likely an object". Imagine the case where there exists a water bottle on an otherwise empty table. While a contextual algorithm would indicate the whole table top as likely region for the presence of a water bottle, a successful object detector would only respond on the water bottle itself. Therefore the two types of methods should not be confused with each other as they are meant for different (but closely related) purposes. We believe that a model which captures 3D structure in the neighborhood of objects can successfully complement object detection algorithms and as we will see, in some cases even outperform them.

An alternative use of the 3D context could also be to suggest good places to put down an object that the robot is carrying. Typically supporting surfaces for objects such as tables or shelves are full of other objects, therefore hindering the performance of finely tuned routines such as plane extraction algorithms. Instead contextual cues can indicate patches of available 3D structure (e.g. an opening on the table surface) where the robot can rest the object it is carrying.

Figure 3.2 shows a scene where the target object is a water tap. The presence of a kitchen sink strongly indicates that the water tap is in its image neighborhood. Parts of the scene that belong to the object's surrounding might be irrelevant to the 3D context and should not be included in the contextual model. During learning the model, we do not know the extent of the neighborhood that is helpful to find the water tap neither the offset between it and the object. We also need a way to represent the 3D structure appropriately. In this work we have opted for a learning based approach where such information can be extracted from data, as opposed to manually connecting certain objects with for example planar surfaces.

In other words, certain regions of the image may predict with high accuracy the presence of an object in another region. In the previous example, the shape of a kitchen sink indicates the presence of a water tap above it. Therefore a model that captures the 3D context of an object needs to explore the object's neighborhood in the scene and find the relevant structures that consistently occur at a certain offset location (if any) with regards to the object.

Another important aspect is that the same object class may appear in different contexts depending on the scene. As an example, a whiteboard marker might appear to be on a vertical surface when it is attached to a whiteboard, however it may also be placed on a table that appear as a horizontal plane. Therefore, we need a model powerful enough to capture a multi-modal context that an object can appear in. We argue that an approach to learn 3D context for certain object classes must capture both the location with respect to the object and the structure of the 3D context. Finally, we find that, while the 3D context captures physically plausible object placements, it can be helped by also including the height. The rationale behind using the height is that for most objects the height at which it appears is quite informative. For example, both trashcans and mugs are often found in regions of space where the surface normals are vertical (the 3D context is a horizontal surface) but the trashcan is on the floor whereas the mug is typically on a table or shelf. This difference is captured by the height.

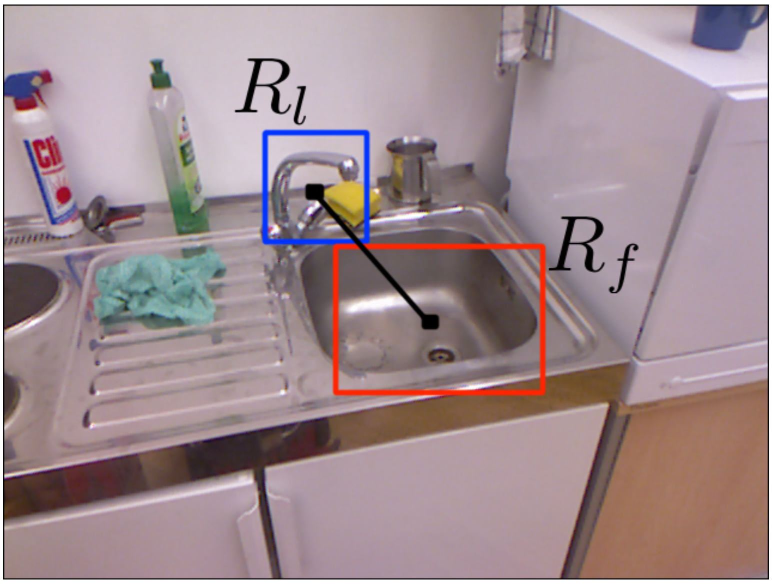


Figure 3.2: The illustration of S_i regions, namely, R_f and R_l and the offset between them.

3.3 Method

In this section, we will present one instantiation of the general ideas presented in Section 3.2. We formulate the problem we address as follows. Given a 3D scene V and an object class O we want find the function

$$f_O(x, y, z, V) = P(\mathrm{X}|V, O) \tag{3.1}$$

This function models how likely it is for an object of class O to be at the location $\mathrm{X} = (x, y, z)$ in the scene. In our approach, we have modeled the function f_O as a binary classifier C_O, where coordinates can be labeled as being a part of the object O or not.

As discussed in Section 3.2, C_O needs to respond to the correct region with respect to the object and needs to be able to handle multi-modal distributions. To account for this, we have chosen to compose C_O from a set of weighted weak binary classifiers $\mathcal{S} = \{S_1, ..., S_i, ...S_n\}$. Each weak classifier models the probability of finding the target object at (x, y, z) given a feature response, F, at an offset location $(x + o_{x,i}, y + o_{y,i}, z + o_{z,i})$. Therefore, each weak classifier explores a specific part of the object's neighborhood in the image, in line with our previous analysis. We have modeled S_i as two rectangular image regions R_f, R_l and the offset between them as shown in Figure 3.2. Here R_f is the region from where a feature response is computed (i.e. the neighborhood hypothesized to be correlated with the object). Further, from R_l we compute a 1D value that corresponds to how object-like R_l is, given by the function $OL(R_l)$. In this work we model $OL(R_l)$ as the fraction of R_l that overlaps with the target object's annotation bounding box [1]. We model 3D context learning as a regression problem, where the goal is, for each S_i to learn the mapping from the feature response of R_f, to $OL(R_l)$.

During training we provide the system with a set of annotated RGB-D images. For each S_i we calculate the feature response in R_f and the objectness measure for the corresponding region $OL(R_l)$. We do this for each object class and every S_i, i.e. combinations of region sizes and offsets. Given this, we then learn which set of the weak classifiers should be utilized[2] to get the final strong classifier C_O. That is, we learn for which S_i there is a strong correlation between a certain feature response in R_f and the object being in R_l. This is formulated as learning the weights for each classifier which effectively results in only a subset of the classifiers being used. Further, using a subset of all the weak classifiers reduces the model complexity and results in a faster processing time. During testing we slide the R_f regions corresponding to the appropriate S_i over the image and get the object responses in the corresponding R_l regions. The response from all active S_i are weighted together according to the learned weights.

3.4 Implementation

In the first step of the training procedure, we slide each S_i across training images. For each, S_i we train a Support Vector Machine (SVM)[3], H_i, with a generalized radial basis function (RBF) kernel to learn aforementioned mapping from a surface patch R_f to the object response in R_l at a certain offset [78]. This results in the set of regressors $\mathcal{H} = \{H_1...H_m\}$. We have chosen to use a discriminative approach since it has been shown that when the amount of labeled training data is large and samples accurately the problem space, discriminative methods tend to work better than their generative counterparts in terms of predictive performance [79, 80, 81]. Furthermore, SVMs being a well understood discriminative method, generally offer lower computational complexity, which is desirable in a robotics context. RBF kernels are shown to provide good results with histogram features [82, 83], which is the feature type used in this work as we will explain later on.

The problem of combining a set of weak regressors to obtain a strong one is well researched in the field. We have chosen the widely used greedy gradient boosting algorithm described in [84] to calculate the vector β which gives the weights with which the output from $\mathcal{H}$ will be scaled. The elements of the weight vector β represents how much each H_i should contribute to the end result. The resulting

[1] Assumed to be available during training.

[2] In the water tap example from before, one S_i might encode the correlation between a sink and a tap behind it and another the correlation between a vertical wall and a tap beneath it.

[3] We have used a modified version of SVM implemented in [77], with $\sigma = 9.7$ and $\epsilon = 0.001$

Table 3.1: The locations from which the dataset is collected and room types from each location.

Site	**Room types**
University of Birmingham	kitchen, classroom, corridor, office, meeting room, computer lab
DFKI - Saarbrücken	corridor, office, meeting room, computer lab
Technical University of Vienna	kitchen, office
University of Ljubljana	kitchen, corridor, office
Royal Institute of Technology	kitchen, corridor, office, meeting room, computer lab

overall regressor is then given by:

$$\mathcal{C}(F(R_f)) = \sum_{i=1}^{n} \beta_i \cdot H_i(F(R_f)) \tag{3.2}$$

where $F(R_f)$ is the feature response for region R_f and $H_i(F(R_f))$ is the i^{th} weak classifier's output. The weighting process results in a large subset of $\mathcal{H}$ having zero or near zero weights. This corresponds to regressors that has low correlation with object presence. The remaining regressors assigned with high weights allows the method to form multi-modal context models as discussed in Section 3.2.

Feature

We would like an expressive feature to represent R_f that is simple and fast to compute. Previous work on 3D features has seen explosive growth in the recent years, due to the availability of 3D sensors and platforms as well as the availability of faster computing [85, 86, 87, 88, 89]. Features proposed in [85, 86, 90] typically are designed for object recognition where the task is towards having a detailed descriptor of a part of an object. The work presented in [87] utilizes both visual and depth values to create a joint feature representation. An interesting feature proposal is Global Structure Histogram by [88] where the idea is to couple a global structure descriptor with local descriptors such as FPFH [85].

As in [69], we refrain from using features that build detailed models of specific 3D shapes, rather we prefer a rough descriptor of a surface patch to capture overall contextual information. For this reason, the feature we have chosen in this work is the point feature histogram, which has been shown to efficiently describe local geometry in 3D point clouds similar to [91]. For a point $\mathbf{p}$ in R_f, a surface normal is computed by fitting a plane to the set of points which are inside a sphere whose center is $\mathbf{p}$ and radius is r. We perform this operation for each pair of points in R_f and obtain a set of vectors as described in [91]. Then these vectors are binned in a 3-dimensional histogram with each dimension containing 8 bins, resulting in 512

Figure 3.3: Example images from the dataset.

bins in total. The height of the center of the object in the scene according to the annotation is concatenated to the feature vector.

3.5 Experiments

The evaluation of a context learner for indoor environments requires a large amount of diverse and real world data. For this reason, we first explain our data set and later on present qualitative and quantitative experimental results.

RGB-D Database

We have constructed our dataset from five different sites in Europe; the Technical University of Vienna (TuV), the University of Birmingham (UB), the Royal Institute of Technology (KTH), the German Center for Artificial Intelligence in Saarbrücken (DFKI) and the University of Ljubljana (UL) (see Figure 3.3). At each site, a Pioneer 3dx robot equipped with a Microsoft Kinect camera tilted down 20° at 1.4 m height was used. The robot is controlled by a human operator using a joystick. It is important to note that the human operators did not know about the purpose of the method presented in this work so as not to bias the collected data. The images were continuously saved as the robot moved through the environment. The dataset can be used for other purposes such as testing 3D mapping and place recognition methods. The data set is available for download from
`http://www.cas.kth.se/rgb-d`.
Table 3.1 details the room types included from each site.

The dataset contains approximately 360GB of Kinect color and depth images. We have annotated five object classes in the dataset:

- cup
- trashcan
- whiteboard marker
- wallplug
- water tap

In total, 1627 images were annotated. The objects in the list were chosen for being frequently found in typical indoor environments and for having diverse context, location and size. As an example, trashcans are on the floor typically near a wall whereas cups are on flat surfaces at a typical table height. Other objects such as whiteboard markers have a less clear contextual one-to-one mapping and can both be on tables and appear to be on a wall. Furthermore, whiteboard markers typically occupy very little space in images in contrast to bigger object such as trashcans. The object water tap has a very distinct 3D context, however the type of scenes it is usually found in are quite cluttered as can be seen from Figure 3.5.

Evaluation

For the evaluation of the method we have selected the KTH, Birmingham and DFKI datasets as the training set and Ljubljana and Vienna as test sets. The training set corresponds to roughly 70% of the images and the remaining images are used for testing.

Qualitative analysis

We will first go through the set of example images and responses shown in Figures 3.5 – 3.13 in order to present a qualitative evaluation of the proposed method.

Figure 3.6 and 3.7 shows two example images for the object *cup*. We can see that the learned context for cups is flat surfaces at the height of a typical table. The method gives high response to these areas as can be seen in figure 3.7 and we can see that the large portions of the image can be ruled out. Interestingly, the method gives a high response for the bottom part of the whiteboard where there's a small flat surface to place whiteboard markers. In this case the method predicts that the small flat surface that is at a similar height of a table can also support a cup.

Figure 3.8, we see that the method learned that a *trashcan* is typically found on the floor but even more specifically at the intersection of a floor and a wall. The high response over the area that corresponds to the object in Figure 3.8 and 3.9 comes from the fact that the sides of the trashcan also appear as a vertical surface on the floor thus resembling a wall.

In Figures 3.10 and 3.11, the two images show that the *whiteboard marker* appears in very distinct contexts, one on the table and the other on the whiteboard, showing that our approach has captured the multi-modal context in which the object appears in the training set. This is captured by our method without the need of a specialized algorithm for each case, such as a table detector and a wall detector as it would be needed in previous work [65]. We note that a large part of the whiteboard is selected by the algorithm, this is due to the variation in average height of which whiteboard markers appear in the dataset.

Results for the object *wallplug* are displayed in Figure 3.12 and 3.13. Similar to whiteboard markers, there are mainly two types of wallplug placements in the dataset: at the intersection of a table and a wall and the other is on a flat wall surface near the floor or at shoulder height. Figures 3.12 and 3.13 show that both contexts are successfully captured by the proposed method. Figure 3.13 is a corridor scene consisting of flat surfaces. In this case, what prevents the method from predicting a high response all over the walls is the height component in the feature vector.

Looking closely at a cluttered scene, Figure 3.5 shows a kitchen scene where the object *water tap* is present. We have picked this object since it has a very complex 3D context which can be disturbed a great deal as a result of clutter such as cups, plates, washing liquid, sponge, dirty dishes. However, certain aspects of its context is very persistent, there is almost always a sink in front of the water tap, which itself has a distinct 3D shape. Another cue is that typically there is a flat wall behind water taps. We can see that the method's highest response has a big overlap with the object itself. An interesting note here is that, the object itself is missing in the depth image, due to its shiny surface. However in this case, since its context is much larger and visible, our method allows point to the object's location even though it is invisible in depth. The area inside the sink, the counter top, the side wall and the oven has lower responses which can be interpreted as they are correctly recognized as the object's 3D context. However we also note that the method gives considerably high responses overall in the image. There are several reasons for this. One is that the high clutter in the training scenes makes it harder to extract the true 3D context of the object. Another is that, using the Kinect camera, scenes containing shiny surfaces result in large amount of noise in depth.

Quantitative analysis

We have evaluated our method quantitatively in three experimental setups to gain statistical insights. First, we have checked how much of the actual object is in the predicted region associated with a specific probability threshold. This tells us how accurate the method is in its location predictions. Second, for varying sizes of the search region represented as the percentage of the image we have computed the overlap of the object's bounding box with this region. This quantifies how much of the search region we can rule out without losing parts of the image that actually contains the object. Third, we have computed the average precision of a state-of-the-art object detection algorithm with and without utilizing 3D context as a pre-processing step in an object detection task.

In the first setup, we have thresholded the response image (here the response image is the visualization of object likelihood as shown in Figure 3) to obtain a binary image with white and black regions. The horizontal axis of Figure 3.5 corresponds to values of this threshold. We compute the overlap between the object's bounding box and its predicted location, i.e. the white regions in the image, which is shown in the vertical axis. The amount of overlap tells us if objects in question typically occur where the algorithm predicts them or not. This overlap is by definition 100% when the threshold is zero, meaning when all of the image is selected. From these results, we can say that intuitively the method performs better for objects that blend with their 3D context. As an example, in depth, the whiteboard marker is almost indistinguishable from the table or the wall itself. This can be used as a complement to object recognition algorithms where an object of the size of whiteboard marker is usually very hard to detect. Furthermore, it is encouraging that even with high threshold values, the majority of the bounding box of cup, whiteboard marker and wallplug is included in the predicted region.

The performance drops faster for bigger objects such as trashcan as the threshold value increases. One reason for this is that these objects are typically observed from a wide variety of viewing angles. The result is that such objects can appear to be in very different backgrounds which are not part of the context. Consider the case where a chair is seen from a top and side view; from the top, it would seem that it is on the floor perhaps near a table, however the side view would show the distant scene in the background unrelated to the local 3D context.

Another reason is that such objects tend to *stick out* of their 3D context. With this we mean that a computer monitor's context consists largely of table surfaces, the average distance of the 3D points on the monitor to the table surface is much greater than for a whiteboard marker. This resulted in a negative bias in the type of evaluation we have selected in this work.

In the second setup, we fix a percentage of the image that is predicted as most likely to contain the object according to the method and check if the object falls in this region. As an example, we pick a number of the pixels that constitutes 10% of the image that are the most likely to contain the object. This selection criteria checks the method's performance for varying gains in efficiency (in this case 90% of the image is eliminated). We can see that, the method can eliminate between 30% - 70% of the scene without losing the target objects. For all object classes, on average at least one third of the image can be omitted without missing the object. This is a promising result indicating that objects are embedded in their 3D context and much can be gained from exploiting it.

In the third and final quantitative test we investigated the effect of exploiting 3D context as a first step in an object detection application using the state of the art object detector presented in [92]. This method builds a part based model for object classes. We have trained each object class using the implementation provided in [93]. After this we have provided two sets of test images to the object detection algorithm: raw images and images that are masked with the thresholded 3D context response (i.e. parts of the image with a too low response is masked

Table 3.2: Object detection results with and without the method presented in this work.

Object label	AveP	AveP with
Cup	0.614	0.813
Whiteboard marker	0.332	0.516
Trashcan	0.541	0.774
Wallplug	0.214	0.519
Watertap	0.221	0.317

out). In order to construct the latter image, we have made an informed choice by looking at the results of the second experimental setup. We have thresholded the image such that the least promising 40% of the test images are omitted from the search region. The rationale behind this is that, as Figure 3.5 shows, we rule out very large portion of the image while still retaining nearly 90% of the object for all classes. This has the effect of eliminating false positives. The average precision results are shown in Table 3.2 where a significant increase in detection performance is observed on all classes. We see that the objects that benefited most are small objects such as whiteboard marker and cup since the texture of these objects cannot be captured sufficiently due to their size and thus are most likely to be mistaken for other parts of the image. These results show that exploiting the 3D context of objects to predict likely locations greatly improves the performance of a state-of-the-art detector for all tested object classes. The object with the least amount of gain from incorporating 3D context is watertap and trashcan. We think this can be explained by the fact that mostly these objects occupy a large portion of the image in the data set which gives less false positives.

In a robotics context, time performance is of crucial importance. The object detection algorithm used in this experimental setup [92] takes on average 23 seconds to compute a detection for one object class. This is prohibitively slow for most robotics applications. The method presented in this work takes between 0.6 to 2.1 seconds to run on a single image of 640x480 resolution on a computer with a 2.26 GHz CPU. Each weak classifier approximately takes 60-80 ms to run. We have chosen the parameter intervals of weak classifiers to cover almost all of the relevant part of the parameter space. The range of window sizes for R_f and R_l are between 10x10 to 100x100 pixels with 10 pixel increments and the offset is chosen in the interval ±50 pixels with 10 pixel increments. This results in 250 weak classifiers trained for each object which covers a very large part of the parameter space for the objects types present in this work. We then also rule out weak classifiers that has a weight lower than 0.01 as they contribute negligibly to the final result.

3.6 Summary and Discussion

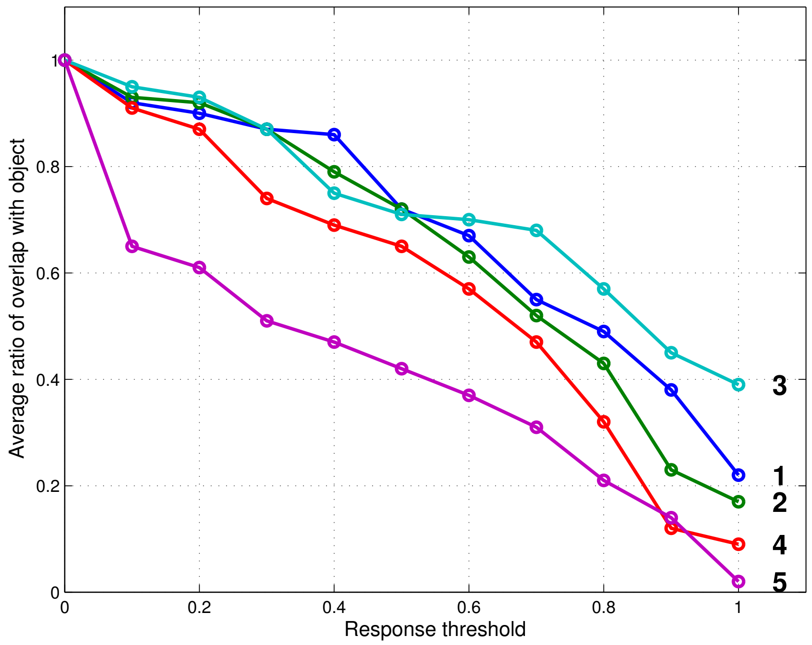
In this work, we have proposed to use local surrounding 3D structure as a strong cue in object placements in everyday scenes, we call this the *3D context* of an object.

We presented a method to extract the 3D context of everyday object and provided extensive quantitative evaluation on a large dataset collected from several real office environments in Europe. Furthermore, we have quantified the benefit of exploiting local structure in an object detection application. The results show that local structure surrounding objects is certainly a strong indicator of object placement in scenes and our method is able to accurately predict the location of the everyday objects included in the study.

It has been shown in previous work that humans possess strong priors about natural scenes [25, 66]. We don't expect objects to float in the middle of a room. We have strong expectations on what type of objects to expect in which scenes and where in the scene. We believe that embodied systems should extract and exploit the structure of the world they perceive. In most mobile robotics systems, a 3D representation of the world is built and maintained for safely navigating the world and manipulating objects. Therefore a robot equipped with a camera should make use of this information when analyzing scenes. Entering a kitchen looking for a cup and being presented with the scene in Figure 3.1, we do not exhaustively scan the whole image including the floor and the walls, instead we almost instantly fixate on the table to get a higher resolution coverage of the table top.

A limitation of all context based approaches is that they are expected to perform poorly in non-typical scenes, where contextual expectations do not agree with the scene at hand. In this case, a global search over the whole image is needed, which is often more expensive than only searching the regions indicated by contextual cues. This is a penalty that is also observed in biological systems [25]. One problem however is that the system needs to detect that a scene is out-of-context, in order to make the decision to perform a global search. The authors in [94] exploit the notion that objects are physically supported by other objects in scenes, similar to the argument used in this work and in our previous work [95, 10].

Open questions includes investigation other methods to capture the 3D context, as with this work we have presented the idea and implemented one instantiation of it. A possible venue is to use 3D context when searching for objects as in [8, 10]. Furthermore, we would like to combine an RGB-D camera with a high resolution photo camera to be able to obtain detailed views of regions predicted by the approach and test the effect of this for object detection. Finally we are interested in employing the 3D context idea in a place categorization framework.



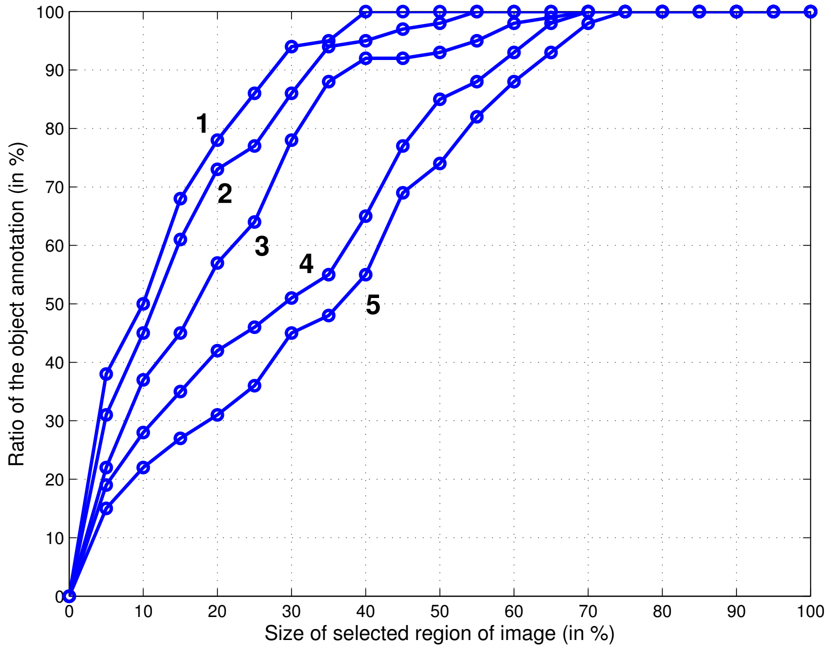


Figure 3.4: Results quantifying the effect of 3D context as a function of (top) probability threshold (bottom) search region size for different objects (1 - cup, 2 - whiteboard marker, 3 - wallplug, 4 - trashcan, 5 - watertap). The top figure shows how much of the object annotation overlaps given a response threshold value. This tells us how much the method's location predictions are in line with the actual object location. The bottom figure shows how much of the image we can eliminate using 3D context while still keeping the actual object bounding box in the remaining part of the image.

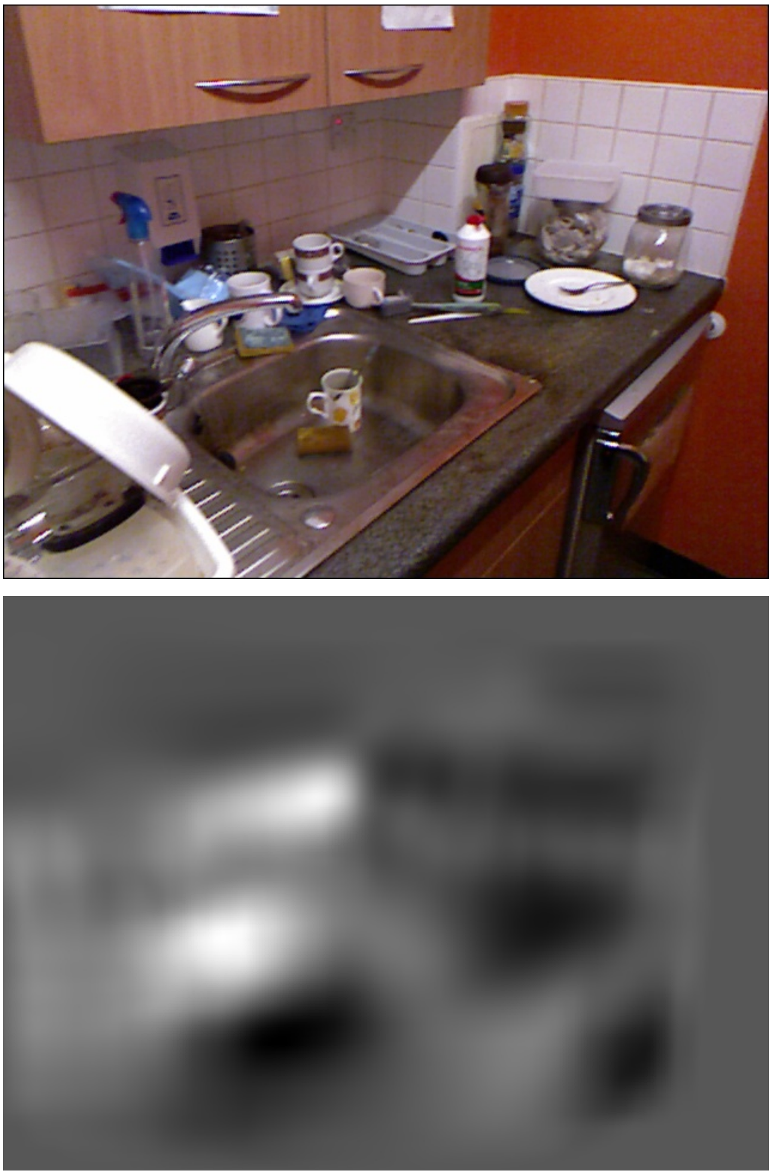


Figure 3.5: An example kitchen scene with a sink and water tap and clutter. The bright areas correspond to higher object presence as predicted by the local 3D context idea. Notice that even though the part of the image corresponding to sink is itself remains dark, the regions above it are computed as promising areas. In this image the water tap object itself is missing in the corresponding depth image due to its specular surface. However since its 3D context is much bigger and has different properties, utilizing it can still point to the object's location.

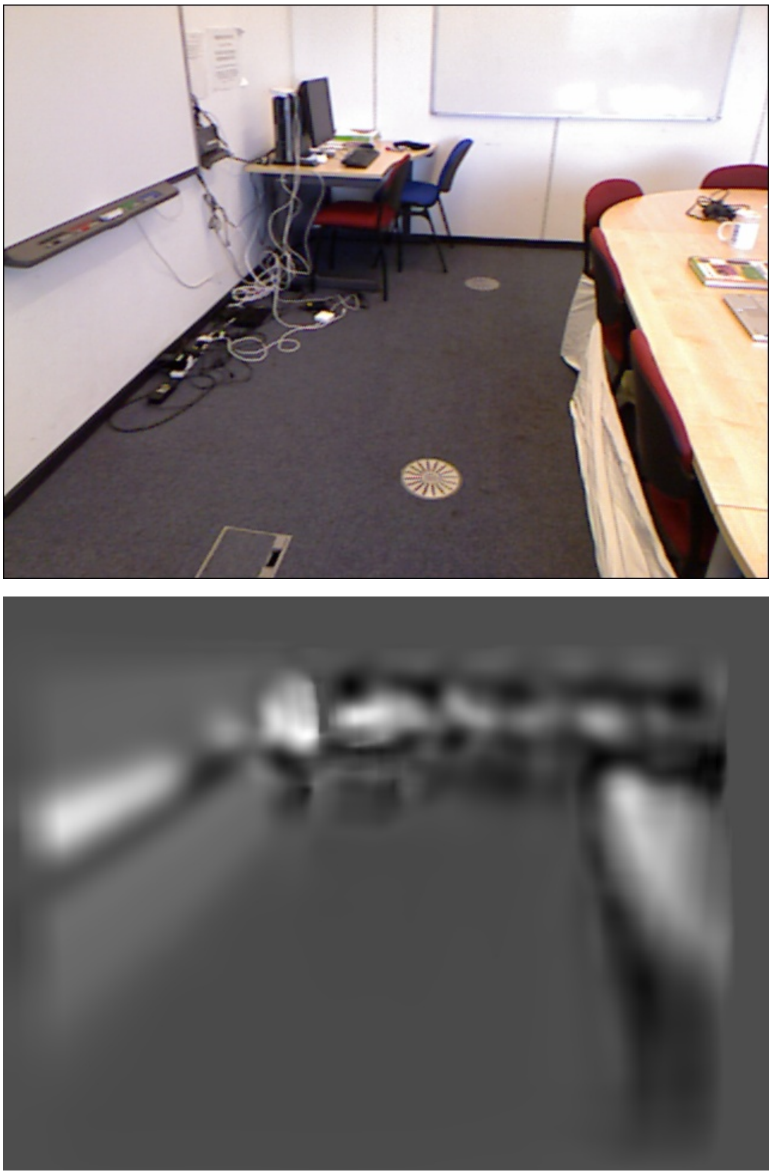


Figure 3.6: Example image from the dataset and responses for the object cup. Brighter areas correspond to a higher likelihood of object presence.

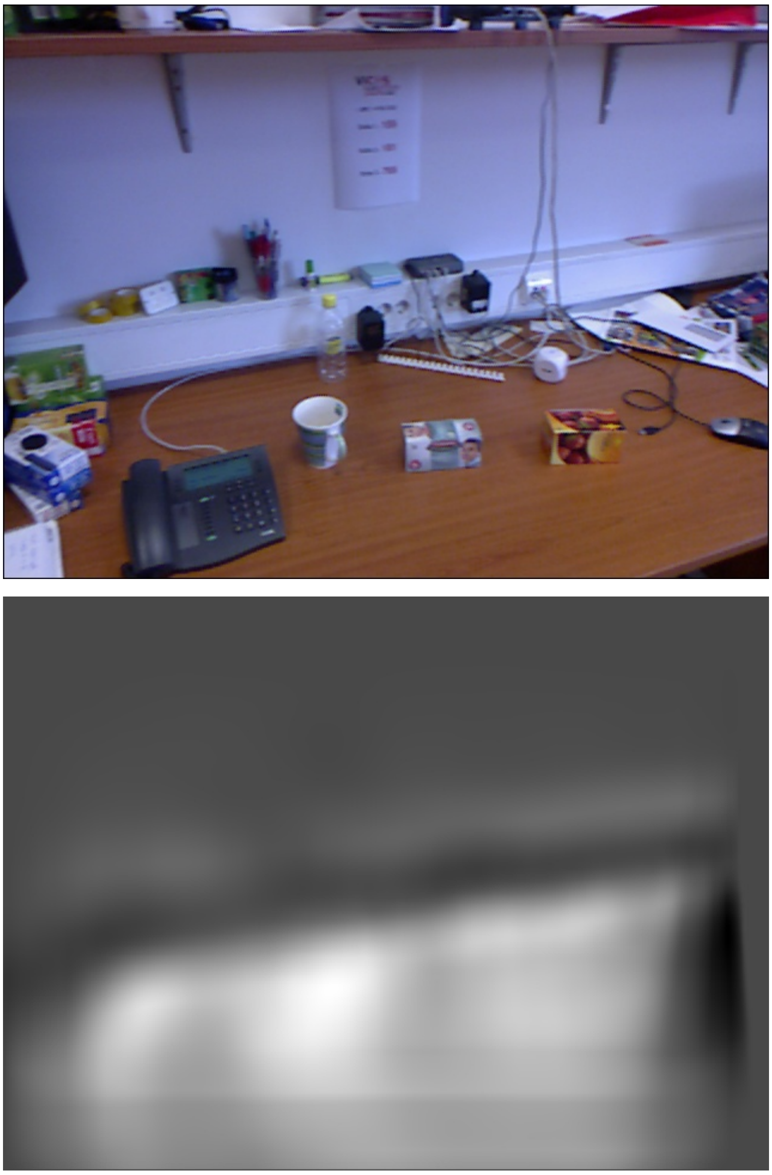


Figure 3.7: Example image from the dataset and responses for the object cup. Brighter areas correspond to a higher likelihood of object presence.

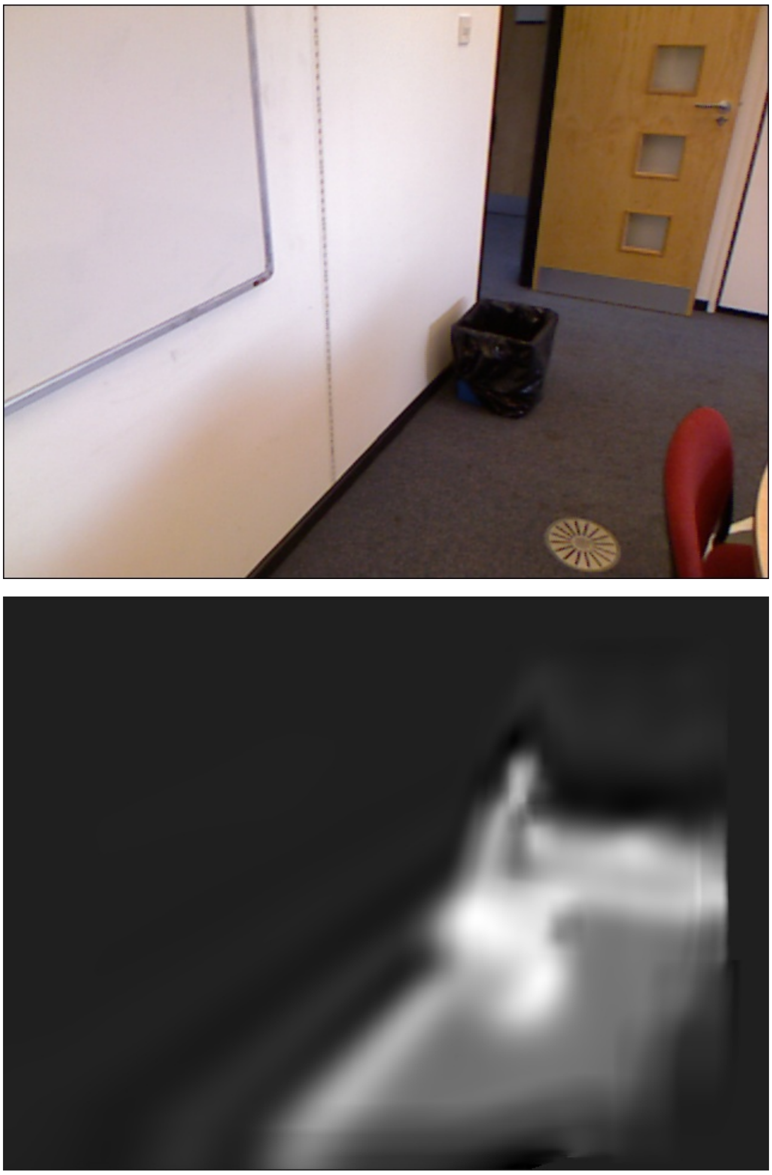


Figure 3.8: Example image from the dataset and responses for the object trashcan. Brighter areas correspond to a higher likelihood of object presence.

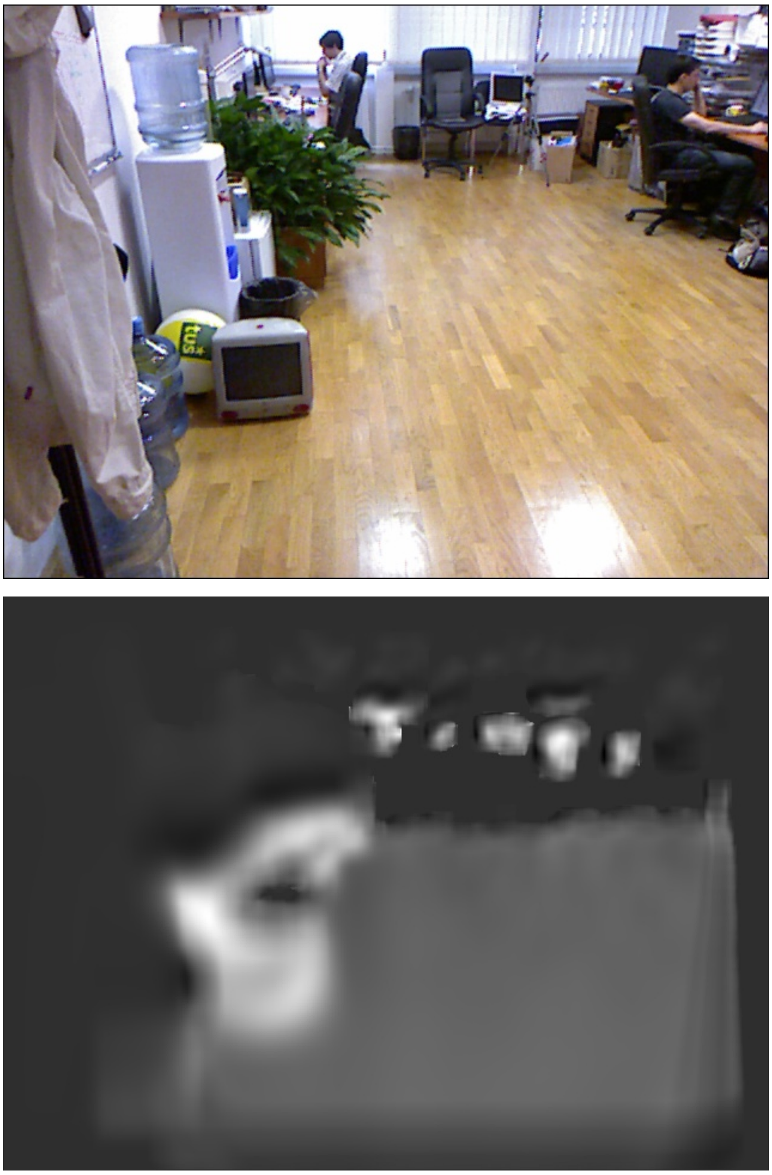


Figure 3.9: Example image from the dataset and responses for the object trashcan. Brighter areas correspond to a higher likelihood of object presence.

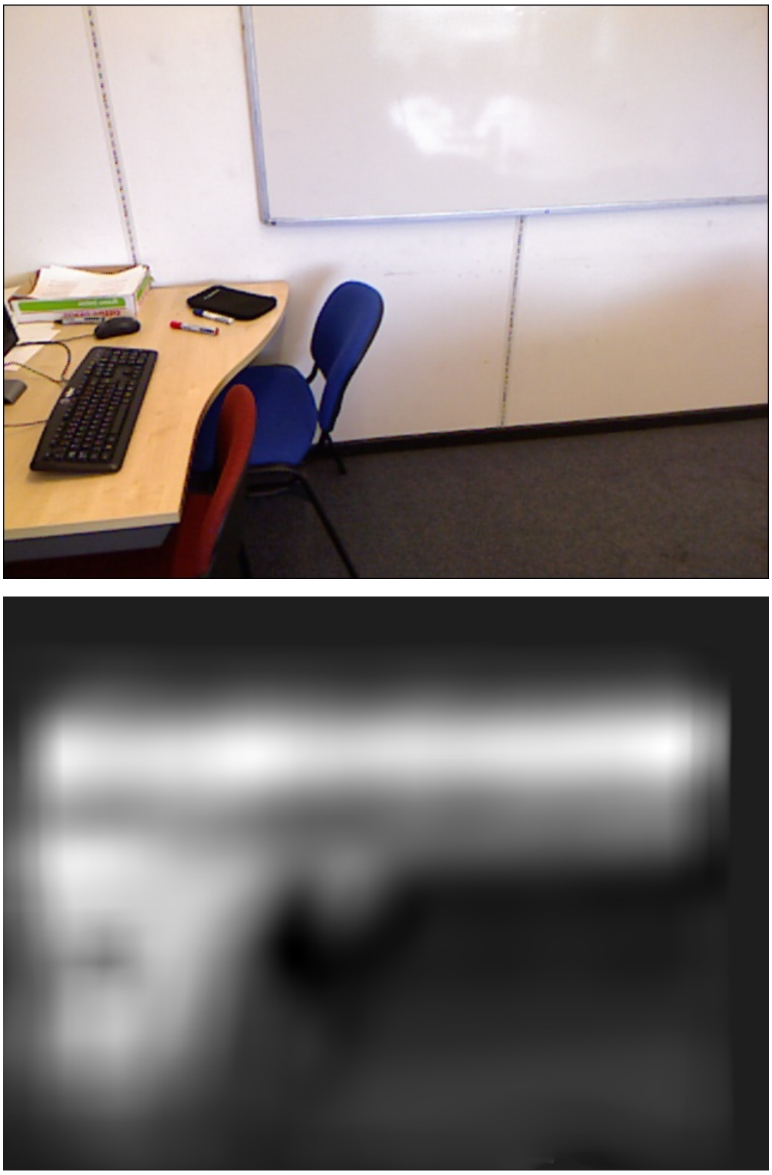


Figure 3.10: Example image from the dataset and responses for the object whiteboard marker. Brighter areas correspond to a higher likelihood of object presence.

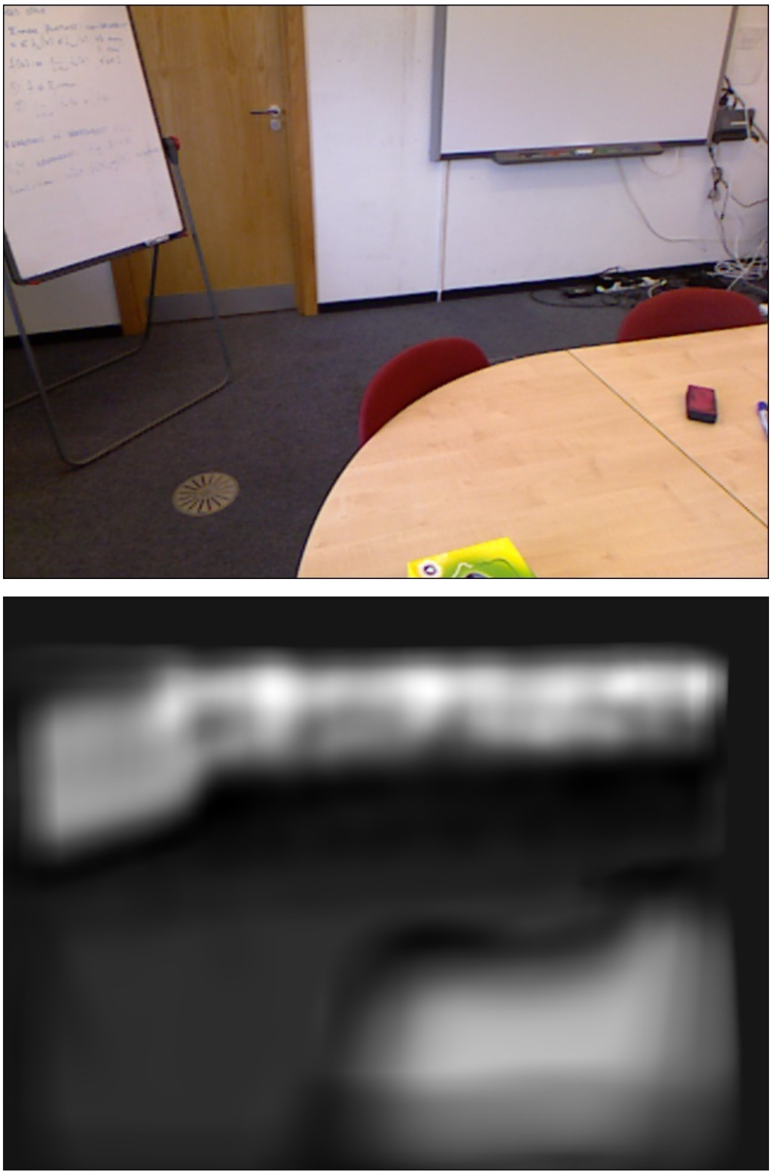


Figure 3.11: Example image from the dataset and responses for the object whiteboard marker. Brighter areas correspond to a higher likelihood of object presence.

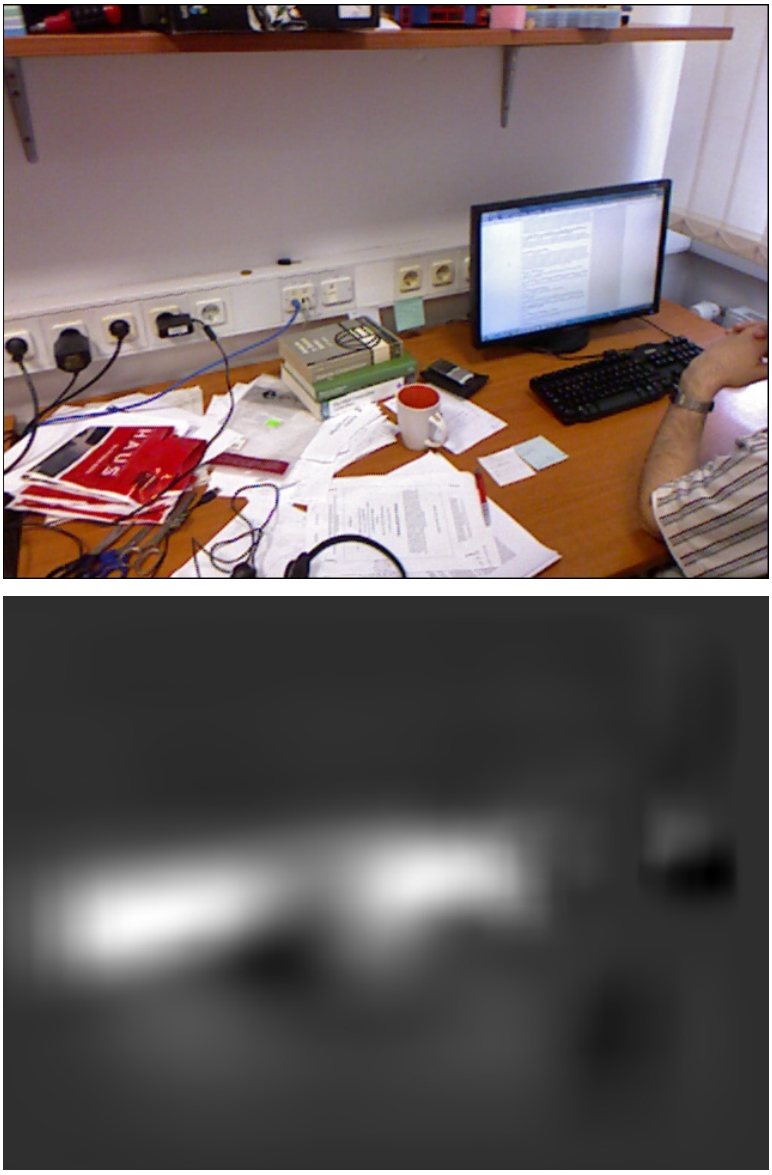


Figure 3.12: Example image from the dataset and responses for the object wallplug. Brighter areas correspond to a higher likelihood of object presence.

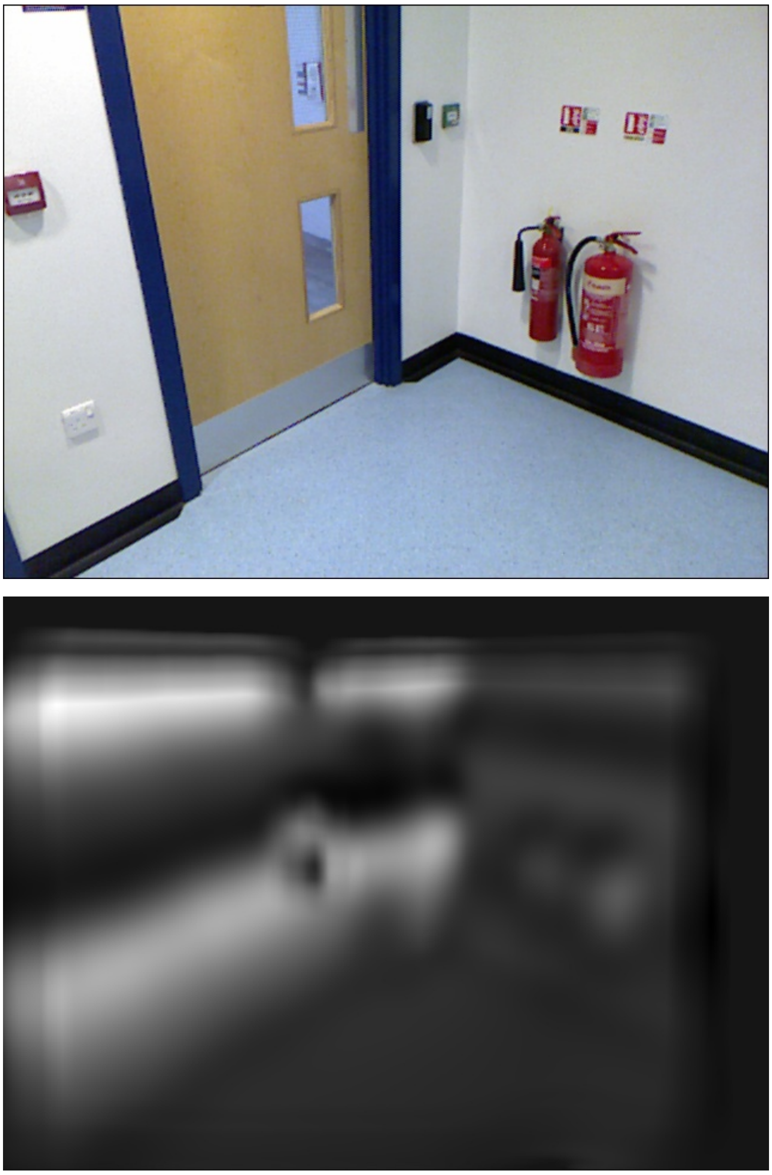


Figure 3.13: Example image from the dataset and responses for the object wallplug. Brighter areas correspond to a higher likelihood of object presence.

Chapter 4

Reasoning about unexplored space in indoor environments using a large floorplan dataset

A large part of robotics research focuses on building autonomous systems which can operate in domestic indoor environments. These works are fueled by the long-term goal of intelligent robots robust enough to handle the complexity of indoor environments. However, as part of this thesis in this chapter, we argue that indoor environments are still not well understood. We do not have a big data view of indoor environments, in contrast to as an example, computer vision where the big data approach recently gained a big interest and proved very useful [96, 97, 98, 99]. Most of previous work incorporates either a few example cases or mock environments, such as a kitchen built in a research lab. One compelling reason for this is that, collecting enough data from actual living spaces to have an accurate sampling of indoors environments at large is a monumental logistics task. This is why most research has stayed limited to using only a few instances of such environments. We note that there are certain efforts in collecting massive indoor datasets such as the upcoming Google Maps Floor Plans, however the research community has not yet taken advantage of such resources [100].

In the previous chapter we have seen an instance of exploiting structure in everyday environments at the scale of single scenes. In this chapter, we will look at indoor environments from a much bigger perspective: at the scale of entire buildings and floorplans. We will analyze a large dataset of annotated floorplans of real buildings, across thousands of actual rooms and hundreds of buildings, orders of magnitude more than found in previous work. We think bringing this type of truly large scale data analysis to robotics will have a large impact and continuation as the field progresses.

In the light of these observations, we have chosen to investigate one particular problem, that is predicting what lies in the unexplored part of the environment.

Reasoning about unexplored space constitutes a crucial part of a mobile robotic system. Various robotics tasks such as rescue missions in unknown buildings or task driven exploration requires the robot to reason about unknown space. The success of a robot's actions in such scenarios depends on the accuracy of the robot's predictions about the unexplored part of the environment.

To illustrate this point, let us revisit the scenario in Chapter 2 and imagine a mobile office robot on the first day of its operation in an unknown office building, tasked with finding an object in an unknown kitchen. The robot needs to plan the set of actions that lead to task completion. In this case, it first needs to find a kitchen by means of exploration. One sensible plan would make use of corridors as connector rooms in most indoor environments. The robot's expectation of finding a kitchen by exploring more of a corridor should be higher than finding a kitchen connected to one of the many other room types it detects during exploration.

Another example is a search-and-rescue robot, entering an unknown office building in the search of survivors. In this case, the robot's goal is to find the maximum amount of survivors in the minimum possible time [101]. The robot, without a complete map of the environment, would need to employ spatial reasoning to estimate what lies in the unknown part of space, prioritizing search actions, which regions to look for survivors in and how best to reach them.

These exemplar scenarios give rise to a set of questions that motivate the work presented in this chapter. How can the robot know the kitchen it's trying to reach is probably reachable (or not) by a corridor? How can it reason that while there may be dozens of office rooms, there's probably one or two kitchens in any given floor? Or that restrooms are typically (but always) at the end of the corridors and not in the middle, so are stairs and elevators? In summary, how can we equip our robots with a basic understanding on the large scale structure of indoor environments?

In this chapter we argue that modeling, extracting and exploiting the structure in indoor environments at the building floor scale can help us in making progress into this problem.

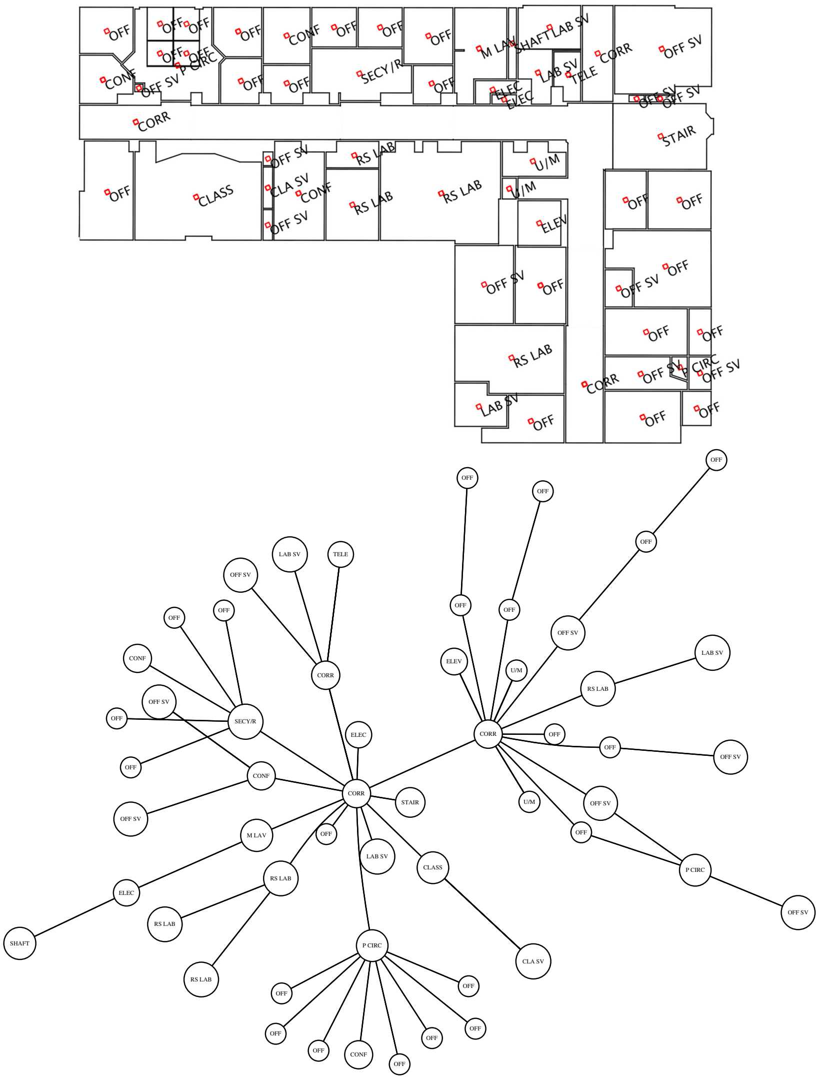


Figure 4.1: Two representations of a building floor. (a) shows the bird-eye view of the geometric layout and (b) the corresponding graph where each room corresponds to a node in the graph and edges between nodes refer to a direct traversable path. Nodes in the graph have labels indicating the category of that room such as kitchen, office, corridor.

As we have seen, high levels of autonomy while operating indoors requires a general understanding of how indoor environments are typically configured, which is currently lacking in state-of-the-art robot systems. In most systems where structural and semantic spatial information is useful, the models of large scale indoor environments are hard-coded and therefore limited [8, 31, 102].

Indoor environments are organized in inter-connected rooms[1]. A widely used representation for this is an undirected graph, where each room corresponds to a node in the graph and edges between nodes refers to a direct path between rooms. Figure 4.1 shows a bird-eye view of a building floorplan represented as a undirected graph. Furthermore, each room can have a label indicating the category of that room, e.g. kitchen, office, corridor, restroom. This type of representation is often called a topological map of the environment.

In this chapter, we adopt a data-driven approach for building models of indoor environments and predicting what lies ahead in the topology of the environment. Therefore, given a partial topological map of a floorplan with room labels, we are interested in *filling in* the rest of the floorplan by predicting the missing parts.

We make use of two large floorplan datasets from the MIT and KTH campuses, with each floor represented as a graph as introduced above. The whole corpus of real world indoor spaces consists of 197 buildings, 940 floors and over 38,000 rooms. To the best of our knowledge no such work exists on data-driven models of indoor maps on this scale. An important property we discover is that, in floorplan graphs local complexity remains nearly constant for increasing global complexity, as we will show, a property observed in other natural graph networks as well. Using this insight, we present two methods for predicting extensions to partial topological maps including room category labels. Finally, we present extensive experimental results on the performance of both methods. In particular, we show how well the models of floorplan graphs learned from the MIT dataset transfer to the KTH dataset.

4.1 Related Work

There exists very limited previous work on reasoning about unexplored space in the topology of environments, also taking into account room categories. However, exploration in unknown environments and reasoning about what lies ahead in the unknown environment has seen a lot of interest since the early days in robotics. We will mention some information gain based. exploration strategies on 2D metric maps. The work by Bourgault et al. is concerned with computing a trajectory that maximizes the accuracy of map building process by a mobile robot [103]. The author's perspective on the problem is improving robot's localization since it is directly tied to the general map quality. The general idea behind the approach is to utilize the uncertainty bounds of the simultaneously built map's landmarks in order to direct the robot towards areas where the uncertainty on robot pose is

[1] Rooms may or may not be separated by physical boundaries, e.g. a kitchen consisting of a sink area and dining area, as recently shown by Sjöö [56]. We make no distinction between the two and refer both as rooms.

highest, in other words, the potential of gaining information is maximum. The authors present experiments on a mobile robot system.

More recently, Stachniss et al. presents an approach that considers exploration, mapping and localization gains while deciding what actions to take in exploring an environment [104]. The significance of this approach is that, by combining all three aspects of mapping an environment, the robot can pick better actions compared to previous methods which only consider one or two of the above as in [103]. The method is based on computing the expected uncertainty of a Rao-Blackwellized particle filter [105] after performing a certain action. Since the space of all possible measurements (and their resulting map and localization updates) after performing an action is prohibitively large, the authors approximates the expected gain. The experiments show that in some cases where an action can both help improving the map accuracy and explore more terrain, the robot choses that action instead of another potential action which only improves the map accuracy.

The work by Vidal-Calleja et al. presents a similar approach to [103] for actively controlling a single hand-held camera for simultaneous localization and mapping (SLAM). Due to real-time computation constraints the authors consider a discrete set of actions. The measure of uncertainty used is the same as [103], that is the entropy of an n-variate Gaussian distribution. However, since it has been shown that using this metric results in non-optimal actions for bearing-only SLAM, the authors proposes to use the Fisher Information Matrix [106].

4.2 Preliminaries and Problem Formulation

We represent each floorplan as an undirected graph. Each node in a graph is assigned a label from an ordered, finite alphabet. The nodes correspond to rooms, the labels are room categories and an edge between two nodes means that there exists a traversable direct path connecting them. An example floorplan graph is shown in Figure 4.1.

A graph is then a three-tuple $G = (V, E, \alpha)$ where V is a finite node set, $E \subseteq V \times V$ is a finite edge set and $\alpha : V \to \mathcal{L}$ is a node label mapping. Let $\mathcal{G}$ be the set of all graphs that can be formed using the label alphabet $\mathcal{L}$.

A *graph database* $\mathcal{D} = \{G_1, ..., G_n\}$ is a set of graphs. Given a graph $G \in \mathcal{G}$ and a graph database $\mathcal{D}$, we define the *projected database* $\mathcal{D}_G$, as the subset of $\mathcal{D}$, for which G is a sub-graph of each element, i.e. $\mathcal{D}_G = \{\forall G' \in \mathcal{D} : G \subseteq G'\}$. The number of elements of the projected database is called the *frequency* of the graph G in the graph database $\mathcal{D}$ and is denoted by $freq(G) = |\mathcal{D}_G|$.

We may now define the *support* of the graph G as:

$$supp(G) = \frac{freq(G)}{|\mathcal{D}|} \tag{4.1}$$

A graph G will be called a *frequent subgraph* in $\mathcal{D}$ if $supp(G) \geq \tau$ where τ is some minimum support threshold, $0 \leq \tau \leq 1$.Finally, we define an *edit operation* to a graph as a node or an edge addition [2].

Problem formulation

The problem statement is the following. Given an incomplete floorplan graph, predict the most likely next room category together with where it is connected to in the current graph or the most likely new path between two rooms.

In formal terms, let G_p be a partial floorplan graph for which the full graph is G. We want to find a certain discrete probability distribution which determines the probability of an edit operation (a node or an edge addition) to G_p. Once this discrete probability distribution is acquired, it is possible to attain the most probable next floorplan graph G'_p. This graph is the result of applying the most probable edit operation upon G_p.

4.3 Method I - Count based prediction

Following the problem formulation, we present the following algorithm. Given an initial input graph G_p, we first compute its projected database $\mathcal{D}_{G_p}$. This means that for each element $G' \in \mathcal{D}_{G_p}$ it holds that $G_p \subseteq G'$. For each edit operation e applicable to G_p with the resulting graph G'_p, we calculate $supp(G'_p)$ as described in Section 4.2. Finally, the edit operation whose resulting graph has the highest support is taken as the most likely node or edge addition to G_p. This method is akin to a hidden Markov Model (HMM) formulation where the state of the model is the graph itself and actions are edit operations.

As we will show in the evaluation section this algorithm performs well for small graph sizes. However, it is limited in the sense that it considers whole graphs at once. Therefore, subtle changes to the input graph can result in drastic changes in the predictions. As an example, if a given input partial graph contains a node with an uncommon or non-existing label in $\mathcal{D}$, the algorithm will only consider those graphs which include this rare node label and disregarding others for building $\mathcal{D}_{G_p}$. Worse yet, in a robotics scenario, it's not unusual for certain rooms remain uncategorized or that the system may be uncertain about the category of a room. In the latter case, method I would report zero probabilities for any possible edit operation. This leads to poor generalization and has the undesired effect of discarding good candidates for prediction.

Therefore we expect that an algorithm that instead exploits the *local structure* in floorplan graphs rather than the entire graph itself would fare better. To develop a better method, we first analyze the properties of floorplan graphs with the goal of finding exploitable structure in the data.

[2] An edge addition should connect two existing nodes in the graph. Likewise, a new node is added with an edge to an existing node.

4.4 Dataset

Each floorplan in the datasets is (stored as an XML file), consists of a set of rooms. Each room has the following:

- The 2D layout represented as line segments and the room's centroid coordinates.
- A set of doorways[3] that indicates a direct traversable path between this room and others it is connected to, if any.
- A category label, e.g. office, corridor, lavatory.

The MIT campus floorplan dataset is originally created by [107] to assist campus travelers by building a machine readable representation of the campus buildings. We propose to use the dataset in a mobile robot context as described in this work. This dataset covers over 160 buildings, 775 floors and nearly 32,000 rooms with 91 categories.

We have created the KTH campus dataset by annotating the architectural floorplan drawings of the KTH campus in order to evaluate our algorithms on a wider set of indoors data and make it available to the community. We have adopted the MIT dataset XML format with the addition that in the KTH dataset, a line segment of a room's layout can be of three types: a wall, a window or a doorway to another room. In the KTH dataset, we have annotated 37 buildings, 165 floors and 6248 rooms. This results in a very rich representation of large scale indoor environments that has not been used in the literature. Next, we will study various properties of both datasets.

4.5 Analysis of the dataset

As discussed earlier, each floorplan can be represented naturally as a graph. The mathematical properties of graphs are well researched. We first start by looking at the global properties of floorplans and then examine the local structure.

Global properties

Watts et al. introduced the property *small-world* to describe the type of graphs in which most nodes can be reached from every other in a small number of steps [108] . Naturally occurring graphs such as social networks between humans and the Internet are shown to be small-world. It is yet to be shown if indoor environments fall into this category. The significance of this in a mobile robot context is that, if floorplans possess small-world characteristics, methods that reason about space can be improved by exploiting its various implications.

[3] A doorway can be either a physical door or a conceptual boundary between two spaces, i.e. printer area in a corridor.

The *clustering coefficient* together with the *characteristic path length* of a graph are used to determine the small-worldness. Graphs that have a high clustering coefficient and low characteristic path compared to that of a random graph[4] can be called small-world graphs. Next we explain both concepts in more detail and present our results.

Characteristic Path Length

We look at the average number of nodes (i.e. rooms) a robot needs to visit in order to travel from one room to another, computed over all pairs of rooms. This is called the characteristic path length of a graph. This value gives insights on how connected indoor environments are, quantitatively the expected amount of rooms to traverse in order to move from one room to another.

The average characteristic path length for increasing graph sizes (i.e. number of rooms in the environment) is shown in Figure 4.2. To elaborate on this, note that for graph sizes between 10 and 25 this value is around 1. What his tells is that, one has to go through on average one another room to be able to reach anywhere on this floor from anywhere else. Our intuition are well in line with this result, this intermediate room usually corresponds to a corridor. The characteristic path length increases roughly linearly with the graph size for larger graphs.

Clustering Coefficient

Widely used in graph analysis, the clustering coefficient, C, indicates whether nodes in a graph are dispersed from each other or form densely connected subgroups. We are interested in this property since it would indicate how locally connected the topology indoor environments are. As an example, figure 4.3 shows two floorplan graphs from the dataset which have different clustering coefficients. The top floorplan in figure 4.3, with a clustering coefficient of 0.56, has multiple connections between neighboring rooms despite having an essentially star shaped topology, whereas all connections in the bottom floorplan goes through a single central room, resulting in a clustering coefficient that is zero. In a mobile robot scenario, if the central room is non-navigable in the bottom floorplan, it would be impossible for the robot go navigate between the rooms, while that is not the case for the top floorplan. This is simply due to the differences in their topologies captured by the clustering coefficient.

We have computed the average clustering coefficient of our database and found it to be $C_{floorplan} = 0.08$. In [108], the authors calculate the clustering coefficient of the United States power grid and the neural network of the worm *C. elegans* among others and found that $C_{powergrid} = 0.08$ and $C_{c.elegans} = 0.28$ respectively, compared to $C_{random} = 0.00027$ and $C_{random} = 0.005$ for random graphs of the same sizes. Comparison to random graphs is important since they are not considered as natural graphs which correspond to man-made networks, such as the topology of

[4] In a random graph, the number of edges that a node has is determined randomly [109].

indoor environments. For more results, [110] gives a comprehensive overview of the results in previous work where the clustering coefficient for several natural graph networks is computed.

Since our floorplan graphs are similar in graph size and sparsity compared to a power grid network and much less dense than a biological neural network graph, it is natural to obtain a value closer to that of the power grid dataset. Another possible conclusion is that similar efficiency concerns are valid for efficient travel of people in the indoors and electricity across the grid. We think analyzing how locally connected rooms are in buildings carries interesting results and key insights, reinforcing the claims made in this chapter.

Local properties

Scale-free graphs

An important local property that is shown to remain invariant in many natural graphs despite high complexity is the scale-free property. In scale-free graphs, the probability that a node has k edges has the following distribution:

$$P(k) \propto k^{-\gamma} \tag{4.2}$$

It has been shown that this seemingly simple property holds for very complex graphs such as the Internet, scientific citations, power grids and social network edges between humans [111, 110]. In effect, this result indicates that across different natural graph datasets, the local connectivity is free of scale and remains unchanged. Figure 4.4 shows the edge distribution of our datasets on logarithmic scale. We have computed γ as 2.073 for graphs of indoor environments compared to 2.1 for the World Wide Web and 2.4 for the United States power grid which are known to possess scale-free characteristics.

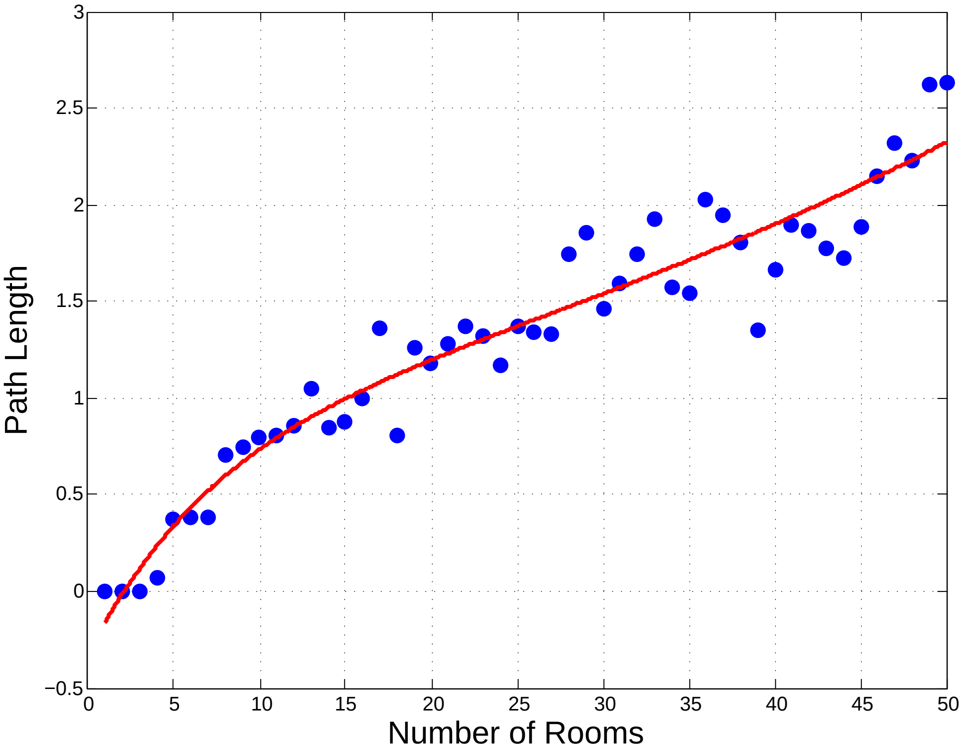


Figure 4.2: Characteristic path length of indoor topological maps for increasing graph sizes.

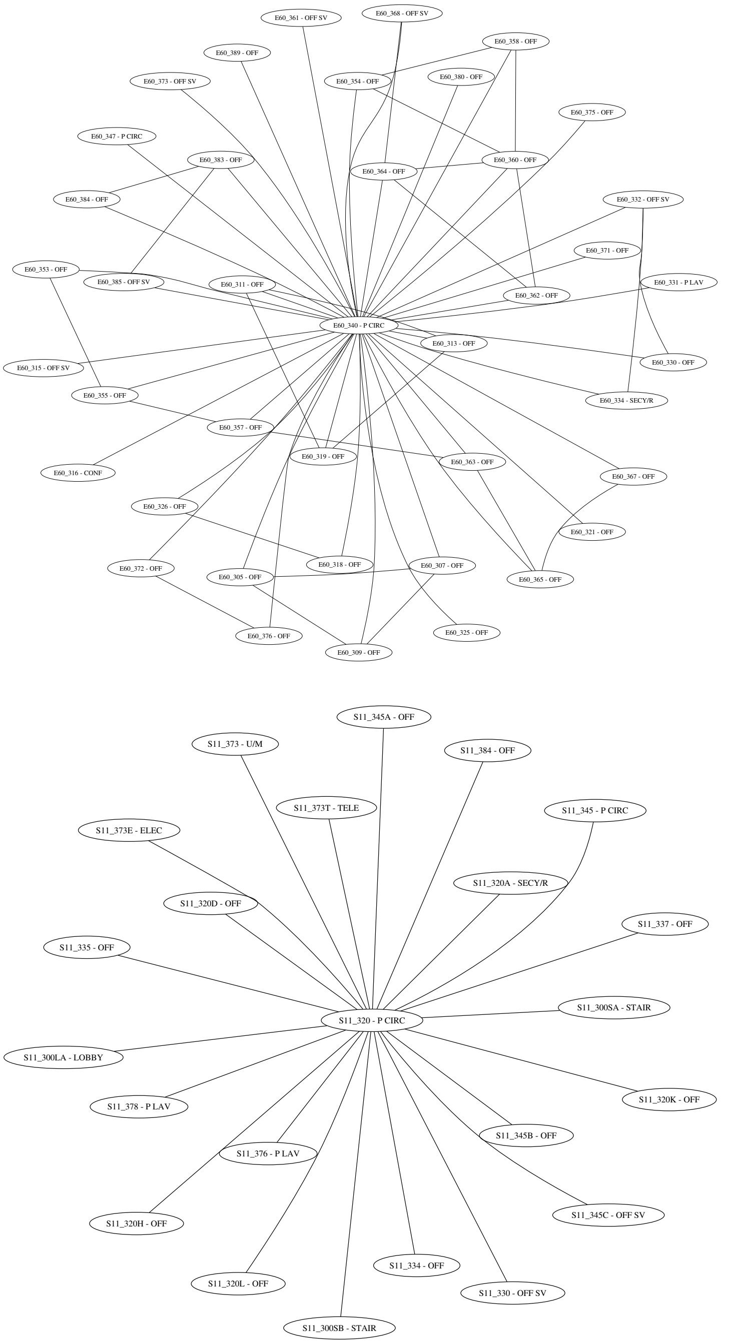


Figure 4.3: Two example floorplan graphs with different clustering coefficients of 0.56 for (a) and 0 for (b). Since certain nodes in (a) have neighbors that are connected to each other, the value for (a) is non-zero, however no such node exists in (b), as a result of this, if the central node is removed, the average path length for the graph in (b) would be infinite.

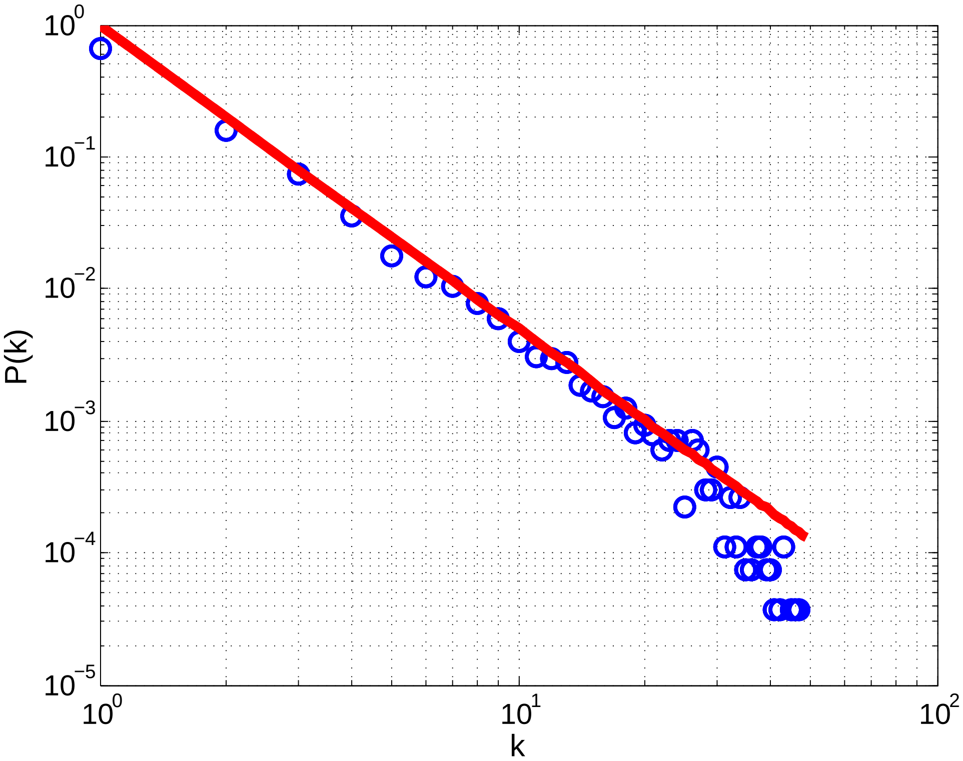


Figure 4.4: This plot shows degree distribution of the dataset in logarithmic scale, namely the probability of a room having a certain k connections to other rooms. The slope of the line gives the γ value discussed in Section 4.5

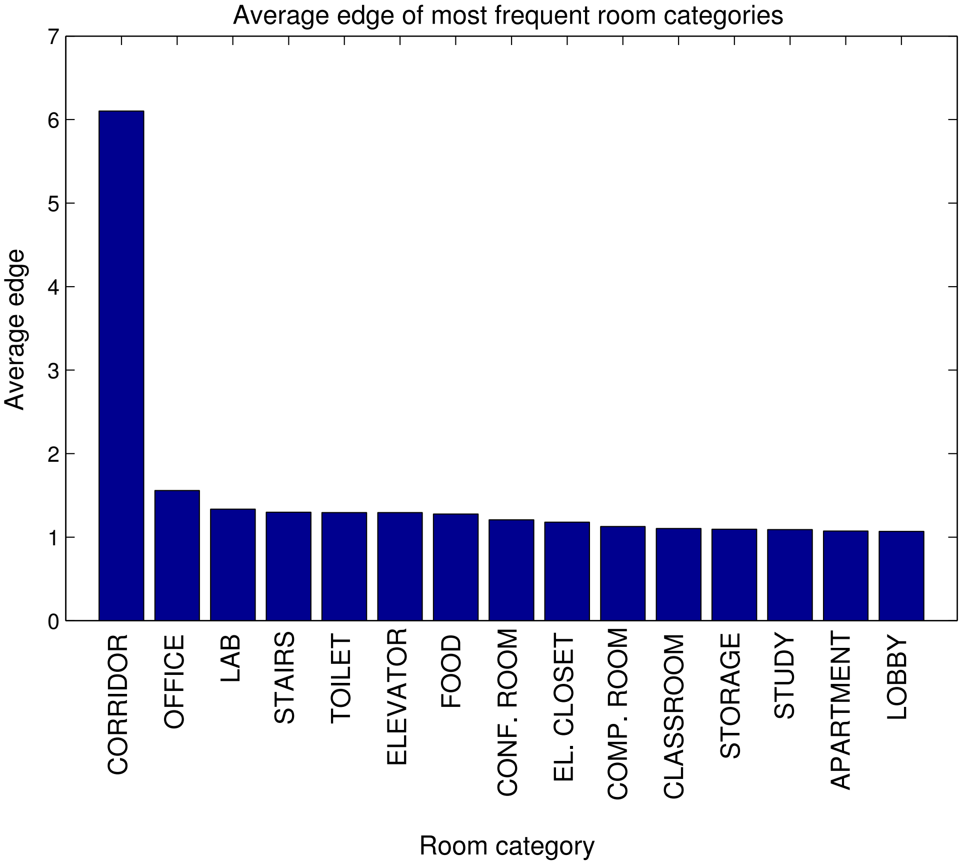


Figure 4.5: Average number of edges for most frequent room categories in both datasets.

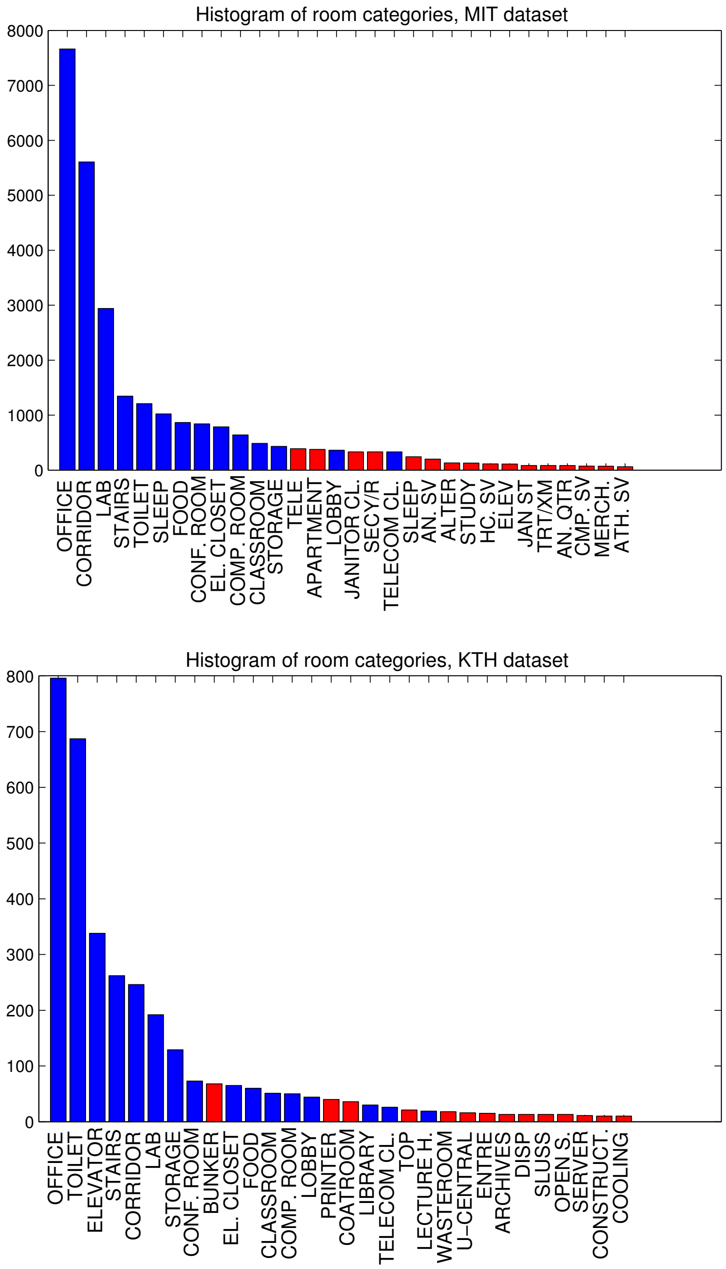


Figure 4.6: Histograms of 30 most frequent categories of the KTH and MIT datasets. The blue color indicates categories that exists on both datasets, red is unmatched categories. Rooms with matched categories constitute 74% and 81% of all rooms in the MIT and KTH datasets respectively.

Single node statistics

Next, we have looked into single node characteristics of floorplan graphs. Figure 4.5 shows the average number of edges per room category, for the 15 most frequent categories in both datasets. Here, we can see that corridors act as connector nodes while the rest of the room categories have much fewer edges.

Figure 4.6 shows most common 30 room categories for the datasets. Here we can see that the general distribution for both datasets are similar with some regional differences. The figure also shows the room categories that both datasets have in common. We see a big overlap there, with 74% and 81% of the categories overlapping for the MIT and KTH datasets, respectively. However, there exists also some big differences. As an example, toilets are much more numerous in the KTH dataset, the reason being is that at KTH, each toilet stall has its own walls and door. We think although the overlap is significant, the differences between the two geographically distant indoor datasets are non-negligible.

Discussion

From this analysis we can see that the topologies of these indoor environments are strongly structured. In particular, a relatively high clustering coefficient indicates that nodes in floorplan graphs tend to form tightly connected subgraphs. In addition, low characteristic path length despite increasing graph sizes together with the edge distribution shown in Figure 4.4 point out that certain rooms have a very high number of direct pathways to other rooms, acting as connector rooms.

With the goal of exploiting this structure in the data, we can argue that a floorplan consists of several functional parts, each of which amounts to a small collection of rooms. As an example, the subgraph $\{office - corridor - stair\}$ is useful for navigation between floors in a building and whereas $\{office - corridor - lavatory\}$ fulfills another necessity in indoor environments. The idea of combining functional parts is also supported by the scale-free property of floorplan graphs as shown previously which displays invariance of local structure for increasing graph sizes. Therefore, we have developed our methods making use of such frequently occurring subgraphs, introduced formally in Section 4.2.

4.6 Method II - Exploiting functional subgraphs

Following the lessons learned from looking at the structure of floorplans, we extract frequently occurring subgraphs from the dataset. The hypothesis put forward in this chapter is that these subgraphs constitute building blocks of the topology of indoor environments. Extracting frequent subgraphs of any size from a graph dataset is known to be an NP-Complete problem. We use the gSpan algorithm [112], which is a fast algorithm for extracting frequently occurring subgraph patterns of any size in a graph dataset. This provides us with the set of frequent subgraphs extracted from our database, $\mathcal{S}$, used in the first step of the method.

The main steps of this method is given in the following:

1. Split the input graph into smaller, overlapping subgraphs, forming a set $\mathcal{C}$.
2. For each subgraph in $\mathcal{C}$, determine the probability of every possible edit operation.
3. Combine the results of the estimates of the edit operations for each subgraph into a final solution for the whole input graph.

Step 1: The goal of this step is to divide the input graph G_p into a set of overlapping connected subgraphs $\mathcal{C} = \{C_1, \ldots, C_m\}$ such that $\forall i \exists j, C_i \cap C_j \neq \emptyset$. The selection of subgraphs C_i plays an important role in the prediction quality. Our hypothesis is that indoor topologies consists of multiple functional smaller parts and we should try to exploit these. We do this by, when possible, picking subgraphs from $\mathcal{S}$ and expand them as viable predictions. Therefore, in the first step of the method, we determine which of the frequent subgraphs from $\mathcal{S}$ that are present in the input graph, and extract the largest possible such frequent subgraphs set. The procedure for computing $\mathcal{C}$ is given in Algorithm 1.

Step 2: In this step, we aim to calculate the probability of all possible edit operations for each element of $\mathcal{C}$. Let $\mathcal{D}_{C_i}$ be the projected database of $C_i \in \mathcal{C}$. Let $\mathcal{X}_{C_i}$ be the set of graphs which are one edit operation away from C_i. We then define, for $x \in \mathcal{X}_{C_i}$, $\phi(x, C_i) = |\{x \subseteq G' \in \mathcal{D}_{C_i}\}|$. This is the number of times we have observed a specific edit operation upon C_i among all the graphs. The most likely edit operation to perform on C_i is then given from $\underset{x \in \mathcal{X}_{C_i}}{\arg\max} \phi(x, C_i)$. This procedure is given in detail in Algorithm 3.

Step 3: The most likely edit operations for each of the subgraphs $C_1, ..., C_m$, lead to new graphs $C'_1, \ldots, C'_m$ respectively. We must select one of these edit operations to get the final prediction outcome. For any selection C'_j made, the resulting prediction will be $G'_p = \bigcup_{i \in [1,m] \setminus \{j\}} C_i \cup C'_j$. We select the edit operation which has the highest support from the graph database. That is, $\underset{C_i, i \in [1,m]}{\arg\max} \phi(C_i, C'_i)$.

Given the function $\phi : \mathcal{G} \times \mathcal{G} \to \mathbb{N}$, it is possible to arrive at an estimate of the discrete probability distribution of the different edit operations upon G_p:

$$P(G'_p = x) = \frac{\phi(x, C_j)}{\sum_{y \in \mathcal{X}_{C_j}} \phi(y, C_j)} \tag{4.3}$$

C_j here refers to the selected subgraph and is chosen as detailed above.

4.7 Experiments

As stated in the problem definition, given a partial graph we want to estimate the probability distribution over possible edit operations, which correspond to predic-

Algorithm 1 Graph splitting

Input:

- G_p, the current partial graph

Output:

- $\mathcal{C} = \{C_1, ..., C_m\}$, the overlapping subgraphs of the partial graph

```
$P \leftarrow \emptyset$
for $s \in \mathcal{S}$ do
  if $s \subseteq G_p \wedge (\neg \exists s' \in \mathcal{S}, s \subseteq s', s' \subseteq G_p)$ then
    $P \leftarrow P \cup \{s\}$
  end if
end for
{$P$ now contains those frequent subgraphs which are contained in the partial
graph $G_p$. They are also the largest possible frequent subgraphs. }
sort(P) by graph size, descending.
$\mathcal{C} \leftarrow \{\text{FindCommonFreqSubgraph}(P, G_p, \emptyset)\}$
while $|G_p| \neq |\bigcup_{i=1}^{n} C_i|$ do
  $Found \leftarrow 0$
  for all $c \in \mathcal{C} \wedge Found = 0$ do
    $c' \leftarrow \text{FindCommonFreqSubgraph}(P, c, \mathcal{C})$
    if $c' \neq \emptyset$ then
      $\mathcal{C} \leftarrow \mathcal{C} \cup c'$
      $Found \leftarrow 1$
      break
    end if
  end for
  if $Found = 0$ then
    $D_g \leftarrow G_p \setminus \bigcup_{i=1}^{n} C_i$
    Add the following vertex set to $D_g$: $\bigcup_{v \in V(D_g)} N(v, G_p) \setminus D_g$
    Add the edges (from the edge set of $G_p$) which correspond to the vertex
    additions above.
    $\mathcal{C} \leftarrow \mathcal{C} \cup \text{GetComponents}(D_g)$
    return $\mathcal{C}$
  end if
end while
return $\mathcal{C}$
```

Algorithm 2 FindCommonFreqSubgraph

This function will attempt to find another frequent subgraph from the set P that has some vertex in common with some graph C_i (the already established subgraphs of G_p).
Input:

- P, the sorted sequence of frequent subgraphs that are present in the partial graph
- G, a graph which the result should have some vertex in common with, this is always some C_i except for the initial execution.
- $C = \{C_1, ..., C_n\}$, the thus far added overlapping subgraphs of the partial graph

Output:

- p, the largest frequent subgraph present in the partial graph that has at least one vertex in common with G (if found). p is also removed from the set P. If no such p could be found, it returns the empty graph $\emptyset$.

```
for all $p \in P$ do
    if HasVertexInCommon$(G, p) \wedge p \not\subseteq \bigcup_{i=1}^{n} C_i$ then
        $P \leftarrow P \setminus \{p\}$
        return p
    end if
end for
return $\emptyset$
```

tions about what lies ahead in the unexplored part of the environment. We have tested the accuracy of predictions for various graph sizes of the input partial graph. The test procedure is as follows. First we pick a random graph G from the dataset and given m, the desired size of the input graph to be included in the test set, we select m connected nodes from G randomly. The m connected nodes and their edges result in subgraph $G_p \subset G$. G_p is added to the test set and this procedure is repeated until the desired amount of test graphs is acquired. Then, for each graph in the test set, we compute the edit operation with the highest probability using both methods. Applying this to G_p results in a new graph G'_p. In order to evaluate the performance, we count every time the new resulting graph G'_p is included in the actual true graph G in the same place as G_p [5]. Note that we train and test on different subsets of the data, so G will never be part of the training data.

We have evaluated both methods on three experimental setups. In the first two experiments, we have trained and tested our algorithm on single datasets. We have

[5] This is to avoid counting the cases where G'_p appears somewhere *else* in G as true positives.

Algorithm 3 Find most likely graph edit operation

Input:

- G, a "small" graph, one subgraph from the output of the graph splitting.
- $\mathcal{D}$, the graph database

Output:

- G', the graph which is the result of performing the optimal edit operation upon G

```
for $x \in \mathcal{D}$ do
  if $G \subseteq x$ then
    for $G' \in B(G, 1) \wedge G' \subseteq x$ do
      {Every $G'$ corresponds to some valid edit operation upon $G$ (that is, both
      $G$ and $G'$ are contained in this specific graph $x$).}
      $\phi(x, G) \leftarrow \phi(x, G) + 1$
    end for
  end if
end for
return $\arg\max_{x \in B(G,1)} \phi(x, G)$
```

used approximately 80% of the datasets for training and the remaining 20% for the testing. In the third experiment, we have used the MIT dataset for training and KTH dataset for testing in order to investigate how transferable the spatial indoor knowledge is between two datasets from distinct places in the world. For all experiments, we have sampled the dataset for 30.000 partial graphs in the test set with no duplicates.

Figure 4.7 shows the results where solid lines denote method I's results and dashed lines correspond to method II's outcome. Furthermore, blue and red lines correspond to the first two experiments where the algorithms are trained and tested on MIT and KTH datasets separately and black lines correspond to the third experiment where we investigate how well the spatial models trained on the MIT dataset transfer to the KTH dataset, we call this the knowledge transfer experiment.

In all cases, we can see that method II performs better compared to method I. Looking more closely, for small graph sizes the difference in performance is minimal since there are a limited number of possible edit operations applicable to input graphs. However, as the graph size increases, i.e. as the graph complexity increases, we can see that method I experiences a sharp drop in performance (blue and red solid lines in Figure 4.7). Earlier, as a result of our analysis in Section 4.5, we hypothesized that indoor topologies consist mostly of smaller, functional parts. We think that method I's performance suffers because identifying such local struc-

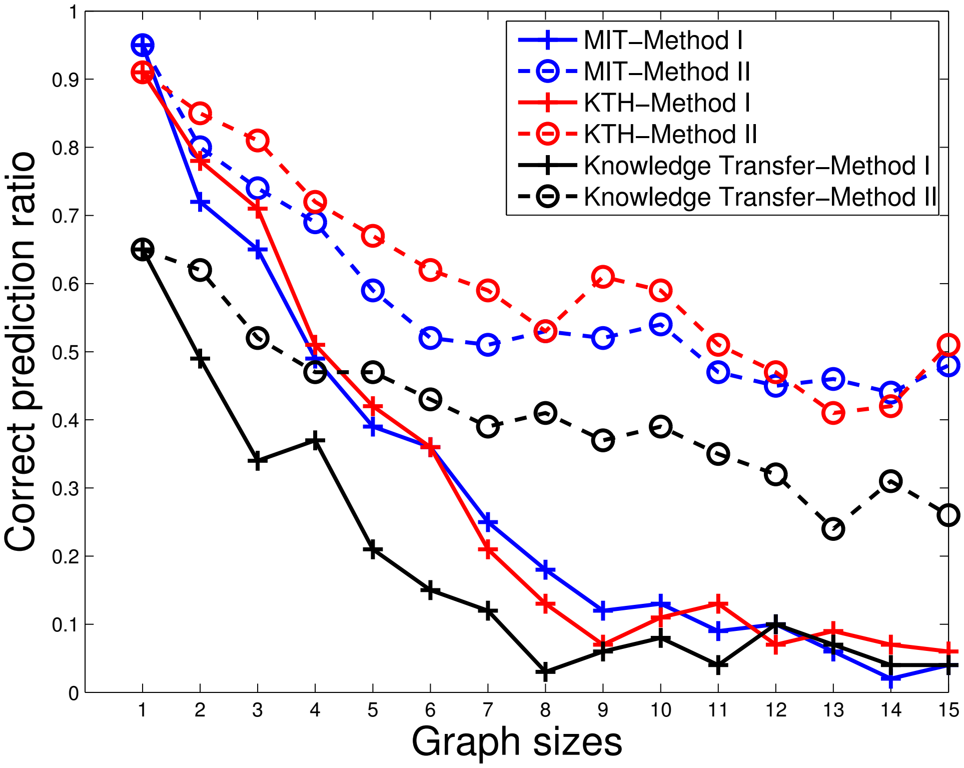


Figure 4.7: The results of the three experimental setups. The solid lines correspond to method I and dashed lines correspond to method II. MIT, KTH and knowledge transfer experiments are color coded blue, red and black respectively.

ture becomes more crucial to make correct predictions as the graph size increases. Method I considers the partial graph monolithically as a whole disregarding any local structure. In comparison, method II identifies frequently occurring parts of graphs and extends those. This makes the latter approach more robust to the artifacts introduced by increasing graph complexity. As an example, if the input graph contains a room with a rarely occurring category, method I would only consider graphs that includes this specific input graph in them, thus ruling out plausible candidates for prediction. This leads to poor generalization of the built spatial model for indoor environment. As opposed to this, method II works on the parts of the graph that are known to be commonplace in the dataset and extends those.

A further observation is that although the method II's performance drops sharply for graph sizes 0-5, the rate of the drop slows down to a great extent for sizes 5-15. Based on our analysis of *scale-free* properties of indoor floorplan graphs laid out in Section 4.5, this behavior may be the result of local complexity in indoor topologies remaining relatively unchanged despite increasing global complexity. By using frequently occurring subgraphs, method II exploits this invariance in local structure in contrast to method I and maintains its correct prediction rate.

In the third experiment, we have evaluated how transferable the knowledge about indoor topologies between the two datasets despite any geographical and cultural biases introduced. First, we analyzed if such a transfer is possible in general. For this reason we have plotted the histogram of room categories of both datasets, and checked how many of each datasets categories overlap with each other. Fig 4.6 displays both histograms where matched categories are shown in blue. The matched categories on both datasets constitute 77% of all rooms in the MIT dataset and 81% in the KTH dataset.

As stated earlier, we trained our algorithms on the MIT dataset and testing on the KTH floorplans. The results are shown as the black lines in Figure 4.7. We can see that the performance suffers compared to single dataset results. One reason for this is that the set of room categories in both datasets are not exactly the same. The algorithm has no way of predicting unmatched categories in the KTH dataset. However despite the apparent unmatched categories and other biases, the results indicated that the models learned from the MIT dataset describes the floorplan graphs of the KTH dataset to a certain extent. Here again, method II performs better compared to method I. An interesting result is that, similar to single dataset experiments, the performance of method II also degrades gracefully for increasing graph sizes compared to method I. We think this is again is due to method II taking advantage of local structure in graphs.

Each prediction takes on average 0.21 seconds to compute on a computer with a single core 2.6 GHz CPU and 8GB of RAM with the implementation not taking advantage of the highly parallelizable characteristic of the algorithms. The frequent subgraphs generated by gSpan algorithm is run offline and cached to file. This takes 62 and 23 seconds for the MIT and KTH datasets respectively, with a frequency threshold of 5%.

4.8 Conclusion and Discussion

Following Chapter 2, in this chapter we have shown that indoor environments are also strongly structured in the large scale, namely at the floorplan level. We have analyzed indoor environments by using two large floorplan datasets. By utilizing tools from graph theory, we have investigated certain statistical properties of indoor environments. An important message from this analysis is that, like many natural graphs, the local structure in indoor topologies remains largely unchanged despite growing global complexity.

We have argued that, the reason for this phenomenon is that indoor environments are organized in functional subunits, therefore even for high overall global complexity, local structure of floorplan graphs remain stable.

As put forward in the introduction chapter of this thesis, we have used the inherent structure of indoor environments this time at a much larger scale, to develop an algorithm that predicts what lies in the unexplored part of the environment. We have compared it to a more straight forward count based method. Prediction capabilities are important for various robotic tasks where the environment is partially known or the robot has uncertain measurements about the environment. We have provided extensive quantitative results on both methods and datasets, also evaluating how well the spatial knowledge can be transferred between two datasets.

We have made the KTH dataset, the annotation tool and an easy to use C++ library for loading and manipulating floorplan databases available at http://www.cas.kth.se/floorplans.

As a continuation, we plan to incorporate more properties to graphs such as geometric properties and objects in order to make more informed predictions. Furthermore, we would like to explore the idea of using machine learning techniques for structured spaces [113], since our output space does not constitute a continuous space of a certain dimension but rather a space of graphs. Exploring different domains such as regular homes and compare their properties to what we have shown in this work is also an interesting research venue.

Chapter 5

Conclusions and Summary

"The robot, equipped by cameras, range sensors and tactile skin feels it all as one great blooming, buzzing confusion..."

Anonymous roboticist, 2012

Autonomous systems which can help humans in everyday environments has been a long term dream of robotics. Amongst the most important obstacles is how to make sense out of the complex and seemingly chaotic world in which robots, imprecisely, perceive and explore their surroundings with various sensors and actuators.

This thesis and the Ph.D. journey behind it focused on investigating how the buzzing confusion of the real world can be distilled into concise and descriptive bits, which can then be presented to autonomous systems, so they can operate more skillfully and robustly in man-made environments. In particular, we were concerned with the orderly nature and structure of man-made environments that these autonomous systems are operating in.

With this in mind, in Chapter 1, we have discussed why it is necessary for researchers to deconstruct this spatial complexity. Briefly, the systems can only perceive the outside world via their sensors, which outputs a stream of numbers, devoid of any higher level concepts such as rooms, furnitures and single objects that we humans have come to use in our reasoning about everyday tasks. We have argued that for a a wide range of applications, understanding and actively exploiting the structure of space is crucial to success. We have outlined what we think are the important questions that need to be answered in discovering, extracting and using this structure.

The questions that arose from Chapter 1 have revealed that this a multi-faceted and broad problem without a clear entry point. In Chapter 2, we addressed a hard to solve application with the goal of revealing more about the questions asked in Chapter 1, and improving state-of-the-art in the said application. We have chosen active visual search as this application, which entails finding objects in the world as its broadest definition. Searching for objects in real world and large environments requires exploiting various facts about the search environment related to spatial structure of the world. We wanted to employ a realistic scenario and search for everyday objects in entire building floors, starting without a map of the environment. We have realized that in order to accomplish this, our systems need to first extract and then exploit the inherent configuration of indoor environments, namely that most objects only appear in certain types of places and rooms . Therefore it is crucial to direct search towards these areas for efficiency and success rate purposes. With this work, we have made a contribution as the realizing the first system which can search for objects in unknown large environments at the scale of entire building floors.

This work has given us the opportunity to experience first hand the difficulties in two areas at different scales, the first, misdetections while finding objects in single images and second, reasoning about unknown part of building floors. In the former, the state-of-the-art vision algorithms that we have used for the most part failed to detect objects in images acquired by a moving robot, especially when those objects appear far away from the camera and there is significant visual artifacts from movement, such as motion blur. The second problem was mainly left untouched in the field, and most reasoning was either done by hand coded routines or researchers has not considered applications where systems had to reason about high-level spatial concepts at the scale of entire building floors.

In line with this, in Chapter 3, we have first investigated the misdetections in single images. The idea here was that, objects are not randomly scattered in the environment, rather, they are typically places at specific locations depending on their purpose and function. We have claimed that this structure of the world can be exploited in order to increase the performance of object detections in a robotics object search scenario. More specifically, we have claimed that the local 3D structure is a strong indication on where objects of certain types can be found in scenes. We have shown one way of capturing the local 3D structure, and performed experiments on a large dataset across multiple object categories.

Moving onto the larger scale, in Chapter 4, we have asked how can robots reason about the unexplored space in indoor environments in order to make decisions such as exploring a corridor in the hopes of finding a kitchen, or looking for another office room since there can be many of them on the current building floor. The goal of this work was to predict the missing parts of an incomplete floor plan by probabilistically predicting what additional rooms and connections between existing rooms there can be. We have amassed a large dataset of annotated floor plans of real buildings and analyzed various statistics of indoor environments at a much larger scale than found in previous work. Based on our analysis of the dataset, we have then designed an algorithm which extracts frequently occurring parts of indoor floor plans in order to make informed predictions about the unexplored part of indoor environments.

In summary, we believe in this thesis we have proposed new ideas on making use of structure in space that has not been explored previously. We have also backed up those ideas with applications to show their plausibility, one can go more deep in each of these areas for further refinement and experimentation.

5.1 Future work

As roboticists, it is our mission to make sense of the confusing buzz of the world for the sake of building ever smart, autonomous robots. We have still a long way to go.

We think that the most important developments in robotics will come from tightening the action-perception loop which currently remains very disconnected, rigid and open-loop. Often, there exists a planner or execution module which works on the symbolic level and which only interacts with what is happening in the world at a level of crude granularity, i.e. when an atomic action (which itself includes various very complex processes) succeeds or fails. We believe for robots to be more responsive, faster and intelligent the effect of the available actions to the system on what the robot perceives should be studied rigorously.

More specifically, we believe the ideas and applications presented in chapters 2, 3 and 4 can provide researchers new research avenues for many years to come and we believe it would be fruitful to pursue them further.

In Chapter 2, the active visual search system presented in this thesis can be improved. This includes improvements to the proposed methods and algorithms in this thesis and other aspects of the problem which we haven't touched yet. One such notable research avenue is better view planning and vision algorithms. Currently the system computes where to look in a rather static way, i.e. first plan to search a room in the environment, then compute where to move the camera and only analyze images taken at those locations. This is not how humans would perform a search and is one of the major causes of performance drop compared to human experiments. Rather at every point in time we analyze all images and make decisions continuously. This ties back with what we said earlier about tightening the action-perception loop.

The work in Chapter 3 on utilizing local geometry in order to predict object locations can be improved by designing different models to capture the 3D structure, both with better 3D point cloud features and better (or different) learning algorithms. We believe the work can also benefit from executing the proposed algorithms on complete maps and integrating it with the active visual search system presented previously, in order to gauge the benefits of utilizing 3D cues in object search. Larger datasets to learn from can also provide the opportunity to test the main idea of Chapter 3 on more object categories and challenging situations.

We think the ideas presented in Chapter 4 can support a very wide range of research questions that has not been explored before in the field. Learning models of indoor environments from very large datasets of annotated building floor plans can help in various areas such as simultaneous localization and mapping, place recognition and categorization, understanding and following route descriptions.

We believe and are content that, this thesis, while providing some answers to the questions it asks, it also opens up new questions that are the subject of research in the years to come, acting as stepping stones to reach the dream of building robots that can live with us in our everyday environments.

Bibliography

[1] L. Von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM, 2004.

[2] E. Law and L. Von Ahn. Input-agreement: a new mechanism for collecting data using human computation games. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 1197–1206, 2009.

[3] Alper Aydemir, Andrzej Pronobis, and Patric Jensfelt. Active visual search in unknown environments using uncertain semantics. *IEEE Transactions in Robotics*, 2012. Conditionally accepted (in review).

[4] Andrzej Pronobis and Patric Jensfelt. Large-scale semantic mapping and reasoning with heterogeneous modalities. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Saint Paul, MN, USA, may 2012.

[5] Moritz Göbelbecker, Charles Gretton, and Richard Dearden. A switching planner for combined task and observation planning. In *Twenty-Fifth Conference on Artificial Intelligence (AAAI)*, August 2011.

[6] A. Aydemir and P. Jensfelt. Exploiting and modeling local 3d structure for predicting object locations. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012.

[7] A. Aydemir, P. Jensfelt, and J. Folkesson. What can we learn from 38,000 rooms? reasoning about unexplored space in indoor environments. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012.

[8] Alper Aydemir, Moritz Göbelbecker, Andrzej Pronobis, Kristoffer Sjöö, and Patric Jensfelt. Plan-based object search and exploration using semantic spatial knowledge in the real world. In *Proc. of the European Conference on Mobile Robotics (ECMR)*, Örebro, Sweden, September 2011.

[9] Alper Aydemir, Kristoffer Sjöö, John Folkesson, and Patric Jensfelt. Search in the real world: Active visual object search based on spatial relations. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.

[10] Kristoffer Sjöö, Alper Aydemir, and Patric Jensfelt. Topological spatial relations for active visual search. *Robotics and Autonomous Systems*, To appear, 2012.

[11] Kristoffer Sjöö, Alper Aydemir, Thomas Mörwald, Kai Zhou, and Patric Jensfelt. Mechanical support as a spatial abstraction for mobile robots. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 2010.

[12] A. Aydemir, D. Henell, P. Jensfelt, and R. Shilkrot. Kinect@home: Crowdsourcing a large 3d dataset of real environments. In *2012 AAAI Spring Symposium Series*, 2012.

[13] Andrzej Pronobis, Kristoffer Sjöö, Alper Aydemir, Adrian N. Bishop, and Patric Jensfelt. A framework for robust cognitive spatial mapping. In *Proc. of the 14th IEEE International Conference on Advanced Robotics (ICAR)*, Munich, Germany, June 2009.

[14] Marc Hanheide, Nick Hawes, Charles Gretton, Alper Aydemir, Hendrik Zender, Andrzej Pronobis, Jeremy Wyatt, and Moritz Gbelbecker. Exploiting probabilistic knowledge under uncertain sensing for efficient robot behaviour. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, Barcelona, Spain, 2011.

[15] Alper Aydemir, Kristoffer Sjöö, and Patric Jensfelt. Object search on a mobile robot using relational spatial information. In *Proc. of the 11th International Conference on Intelligent Autonomous Systems (IAS)*, Ottawa, Canada, August 2010.

[16] Andrzej Pronobis, Kristoffer Sjöö, Alper Aydemir, Adrian N. Bishop, and Patric Jensfelt. Representing spatial knowledge in mobile cognitive systems. In *Proc. of the 11th International Conference on Intelligent Autonomous Systems (IAS)*, Ottawa, Canada, August 2010.

[17] Shrihari Vasudevan, Stefan Gächter, and Roland Siegwart. Cognitive spatial representations for mobile robots - perspectives from a user study. In *Proc. of the Workshop "Semantic information in robotics" at the IEEE International Conference on Robotics and Automation (ICRA'07)*, 2007.

[18] Yiming Ye. *Sensor planning for object search.* PhD thesis, University of Toronto, 1998.

[19] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966 –1005, aug 1988.

[20] John K. Tsotsos. On the relative complexity of active vs. passive visual search. *International Journal of Computer Vision*, 7(2):127–141, 1992.

[21] John K. Tsotsos Yiming Ye. A complexity-level analysis of the sensor planning task for object search. *Computational Intelligence*, 17(4), 2001.

[22] Thomas D. Garvey. Perceptual strategies for purposive vision. Technical Report 117, AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, Sep 1976.

[23] Lambert E. Wixson and Dana H. Ballard. Using intermediate objects to improve the efficiency of visual search. *International Journal of Computer Vision*, 12(2-3):209–230, 1994.

[24] Irving Biederman, Robert J. Mezzanotte, and Jan C. Rabinowitz. Scene perception: Detecting and judging objects undergoing relational violations. *Cognitive Psychology*, 14(2):143–177, April 1982.

[25] Biederman. On the semantics of a glance at a scene. In *Perceptual organization*, pages 213–263. Lawrence Erlbaum Publisher, 1981.

[26] Yiming Ye and John K. Tsotsos. Sensor planning for 3d object search. *Computer Vision and Image Understanding*, 73(2):145–168, 1999.

[27] Ksenia Shubina and John K. Tsotsos. Visual search for an object in a 3d environment using a mobile robot. *Computer Vision and Image Understanding, Special issue on Intelligent Vision Systems*, 114(5):535 – 547, 2010.

[28] A. Andreopoulos, S. Hasler, H. Wersing, H. Janssen, J.K. Tsotsos, and E. Korner. Active 3d object localization using a humanoid robot. *IEEE Transactions on Robotics*, 27(1):47 –64, Feb 2011.

[29] Jeremy Ma, Timothy H Chung, and Joel Burdick. A probabilistic framework for object search with 6-DOF pose estimation. *The International Journal of Robotics Research*, 30(10):1209–1228, 2011.

[30] Staffan Ekvall, Danica Kragic, and Patric Jensfelt. Object detection and mapping for service robot tasks. *Robotica: International Journal of Information, Education and Research in Robotics and Artificial Intelligence*, 2007.

[31] Thomas Kollar and Nicholas Roy. Utilizing object-object and object-scene context when planning to find things. In *Proceedings of the IEEE international conference on Robotics and Automation (ICRA)*, pages 4116–4121, Piscataway, NJ, USA, 2009. IEEE Press.

[32] Per-Erik Forssén, David Meger, Kevin Lai, Scott Helmer, James J. Little, and David G. Lowe. Informed visual search: Combining attention and object recognition. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 935–942, Pasadena, CA, USA, May 2008. IEEE, IEEE Robotics and Automation Society.

[33] Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, june 2007.

[34] Javier Velez, Garrett Hemann, Albert S. Huang, Ingmar Posner, and Nicholas Roy. Modelling observation correlations for active exploration and robust object detection. *Journal of Artificial Intelligence Research*, 44:423–453, July 2012.

[35] P. Viswanathan, D. Meger, T. Southey, J.J. Little, and A.K. Mackworth. Automated spatial-semantic modeling with applications to place labeling and informed search. In *Canadian Conference on Computer and Robot Vision (CRV)*, pages 284 –291, may. 2009.

[36] Sumantra Dutta Roy, Santanu Chaudhury, and Subhashis Banerjee. Active recognition through next view planning: a survey. *Pattern Recognition*, 37(3):429 – 446, 2004.

[37] J. O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, New York, NY, 1987.

[38] T. C. Shermer. Recent results in art galleries [geometry]. *Proceedings of the IEEE*, 80(9):1384–1399, Sep 1992.

[39] B. J. Nilsson. *Guarding Art Galleries — Methods for Mobile Guards.* PhD thesis, Lund University, 1995.

[40] H. González-Banos. A randomized art-gallery algorithm for sensor placement. In *Proceedings of the seventeenth annual symposium on Computational geometry*, pages 232–240, New York, NY, USA, 2001. ACM.

[41] S.M. LaValle, D. Lin, L.J. Guibas, J.-C. Latombe, and R. Motwani. Finding an unpredictable target in a workspace with obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 737 –742 vol.1, apr 1997.

[42] A. Sarmiento, R. Murrieta, and S.A. Hutchinson. An efficient strategy for rapidly finding an object in a polygonal world. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 1153 – 1158 vol.2, oct. 2003.

[43] A. Sarmiento, R. Murrieta-Cidz, and S. Hutchinson. A Sample-based Convex Cover for Rapidly Finding an Object in a 3-D Environment. pages 3486–3491, April 2005.

[44] Geoffrey Hollinger, Dave Ferguson, Siddhartha Srinivasa, and Sanjiv Singh. Combining search and action for mobile robots. In *Proceedings of the IEEE International conference on Robotics and Automation (ICRA)*, pages 800–805, Piscataway, NJ, USA, 2009. IEEE Press.

[45] H. Masuzawa and J. Miura. Observation planning for efficient environment information summarization. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5794 –5800, oct. 2009.

[46] Mathieu Boussard and Jun Miura. Object Search: A Constrained MDP Approach. In *Proceedings of the IEEE/RSJ international conference on Intelligent robots and systems (IROS), Workshop on Active Perception and Object Search in the Real World*, CA, USA, 2011.

[47] Dominik Joho, Martin Senk, and Wolfram Burgard. Learning search heuristics for finding objects in structured environments. *Robotics and Autonomous Systems*, 59(5):319–328, May 2011.

[48] Travis Deyle, Hai Nguyen, Matt Reynolds, and Charles C. Kemp. RF vision: RFID receive signal strength indicator (RSSI) images for sensor fusion and mobile manipulation. In *Proceedings of the IEEE/RSJ international conference on Intelligent robots and systems (IROS)*, pages 5553–5560, Piscataway, NJ, USA, 2009. IEEE Press.

[49] Mehdi Samadi, Thomas Kollar, and Manuela M. Veloso. Using the web to interactively learn to find objects. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, July 2012.

[50] Thomas Kollar, Mehdi Samadi, and Manuela Veloso. Enabling robots to find and fetch objects by querying the web. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 3*, AAMAS '12, pages 1217–1218, Richland, SC, 2012. International Foundation for Autonomous Agents and Multiagent Systems.

[51] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth J. Teller, and Nicholas Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[52] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth J. Teller, and Nicholas Roy. Approaching the

symbol grounding problem with probabilistic graphical models. *AI Magazine*, 32(4):64–76, 2011.

[53] Lars Kunze, Michael Beetz, Manabu Saito, Haseru Azuma, Kei Okada, and Masayuki Inaba. Searching objects in large-scale indoor environments: A decision-thereotic approach. In *IEEE International Conference on Robotics and Automation (ICRA)*, St. Paul, MN, USA, May 14–18 2012.

[54] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 101(1-2):99–134, May 1998.

[55] Kristoffer Sjöö, Dorian Gálvez López, Chandana Paul, Patric Jensfelt, and Danica Kragic. Object search and localization for an indoor mobile robot. *Journal of Computing and Information Technology*, 17(1):67–80, 2009.

[56] Kristoffer Sjöö. Semantic map segmentation using function-based energy maximization. *IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[57] S. Vasudevan and R. Siegwart. Bayesian space conceptualization and place classification for semantic maps in mobile robotics. *Robotics and Autonomous Systems*, 56:522–537, June 2008.

[58] Hendrik Zender, Óscar Martínez Mozos, Patric Jensfelt, Geert-Jan M. Kruijff, and Wolfram Burgard. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6):493–502, June 2008.

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

[60] Jeremy L Wyatt, Alper Aydemir, Michael Brenner, Marc Hanheide, Nick Hawes, Patric Jensfelt, Matej Kristan, Geert-Jan M Kruijff, Pierre Lison, Andrzej Pronobis, Kristoffer Sjöö, Danijel Skočaj Alen Vrečko, Hendrik Zender, and Michael Zillich. Self-understanding and self-extension: a systems and representational approach. *IEEE Transactions on Autonomous Mental Development*, 2(4):282–303, 2010.

[61] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 146–151, Monterey, CA, July 1997.

[62] Steffen L. Lauritzen and Thomas S. Richardson. Chain graph models and their causal interpretations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(3):321–348, 2002.

[63] Marc Hanheide, Charles Gretton, Richard W. Dearden, Nick A. Hawes, Jeremy L. Wyatt, Andrzej Pronobis, Alper Aydemir, Moritz Göbelbecker, and Hendrik Zender. Exploiting probabilistic knowledge under uncertain sensing for efficient robot behaviour. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, Barcelona, Spain, jul 2011.

[64] Kristoffer Sjöö. *Functional understanding of space : Representing spatial knowledge using concepts grounded in an agent's purpose.* PhD thesis, Royal Institute of Technology (KTH), 2011.

[65] Zoltan Csaba Marton, Radu Bogdan Rusu, Dominik Jain, Ulrich Klank, and Michael Beetz. Probabilistic categorization of kitchen objects in table settings with a composite sensor. In *The 22nd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, 10/2009 2009.

[66] Moshe Bar. Visual objects in context. *Nature Reviews: Neuroscience*, 5(8):617–629, August 2004.

[67] Barbara Hidalgo-sotelo, Aude Oliva, and Antonio Torralba. Human learning of contextual priors for object search: Where does the time go. In *Proceedings of the 3rd Workshop on Attention and Performance in Computer Vision*, 2005.

[68] Antonio Torralba, Monica S. Castelhano, John M. Henderson, and Aude Oliva. Contextual guidance of eye movements and attention in real-world scenes: The role of global features in object search. *Psychological Review*, 113(4):766–786, 2006.

[69] Antonio Torralba. Contextual priming for object detection. *Int. J. Comput. Vision*, 53:169–191, July 2003.

[70] Antonio Torralba, Aude Oliva, Monica S. Castelhano, and John M. Henderson. Saliency, objects and scenes: global scene factors in attention and object detection. *Journal of Vision*, 4(8):337, 2004.

[71] Dejan Pangercic, Moritz Tenorth, Dominik Jain, and Michael Beetz. Combining perception and knowledge processing for everyday manipulation. In *in Proc. of IEEE/RSJ International Conference on Intelligent RObots and Systems (IROS'10)*, 2010.

[72] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. PAMI*, 20(11):1254–1259, 1998.

[73] Simone Frintrop. *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search.* PhD thesis, University of Bonn, July 2005.

[74] Simone Frintrop and Patric Jensfelt. Attentional landmarks and active gaze control for visual SLAM. *IEEE Transactions on Robotics, special Issue on Visual SLAM*, 24(5), October 2008.

[75] Mårten Björkman and Jan-Olof Eklundh. Vision in the real world: Finding, attending and recognizing objects. *International Journal of Imaging Systems and Technology*, 16(5):189–208, 2006.

[76] S. Frintrop, A. Nuchter, H. Surmann, and J. Hertzberg. Saliency-based object recognition in 3d data. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2167 – 2172 vol.3, sept.-2 oct. 2004.

[77] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2:27:1–27:27, May 2011.

[78] Bernhard Schlkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik. Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing*, 45:2758–2765, 1997.

[79] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *NIPS*, pages 841–848, 2001.

[80] J.A. Lasserre, C.M. Bishop, and T.P. Minka. Principled hybrids of generative and discriminative models. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 87 – 94, june 2006.

[81] Thomas Deselaers, Georg Heigold, and Hermann Ney. Object classification by fusing svms and gaussian mixtures. *Pattern Recognition*, 43(7):2476–2484, 2010.

[82] V. Sreekanth, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Generalized RBF feature maps for efficient detection. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2010.

[83] O. Chapelle, P. Haffner, and V.N. Vapnik. Support vector machines for histogram-based image classification. *Neural Networks, IEEE Transactions on*, 10(5):1055 –1064, sep 1999.

[84] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[85] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *The IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 05/2009 2009.

[86] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, and John Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 10/2010 2010.

[87] Asako Kanezaki, Takahiro Suzuki, Tatsuya Harada, and Yasuo Kuniyoshi. Fast object detection for robots in a cluttered indoor environment using integral 3d feature table. In *ICRA*, pages 4026–4033. IEEE press, 2011.

[88] Marianna Madry, Carl Henrik Ek, Renaud Detry, Kaiyu Hang, and Danica Kragic. Improving Generalization for 3D Object Categorization with Global Structure Histograms. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2012. to appear.

[89] Torsten Fiolka, Jörg Stückler, Dominik Alexander Klein, Dirk Schulz, and Sven Behnke. Sure: Surface entropy for distinctive 3d features. In *Spatial Cognition*, pages 74–93, 2012.

[90] Bastian Steder, Radu Bogdan Rusu, Kurt Konolige, and Wolfram Burgard. Narf: 3d range image features for object recognition. In *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 8, 2010 2010.

[91] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Aligning point cloud views using persistent feature histograms. In *Proceedings of the 21st IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, 2008 2008.

[92] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, September 2010.

[93] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. http://people.cs.uchicago.edu/ pff/latent-release4/.

[94] Alan S. Willsky Myung Jin Choi, Antonio Torralba. Context models and out-of-context objects. *Pattern Recognition Letters*, to appear, 2012.

[95] Kristoffer Sjöö, Alper Aydemir, Thomas Mörwald, Kai Zho, and Patric Jensfelt. Mechanical support as a spatial abstraction for mobile robots. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'10)*, October 2010.

[96] Jia Deng, Samy Bengio, Yuanqing Lin, and Fei-Fei Li. Large scale learning for vision. In *Workshop at Computer Vision and Pattern Recognition (CVPR) 2011*, 2011.

[97] Alex Berg and Rob Fergus. Big data meets computer vision. In *First International Workshop on Large Scale Visual Recognition and Retrieval at Neural Information Processing Systems*, 2012.

[98] A. Torralba, R. Fergus, and W.T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11):1958–1970, 2008.

[99] Q.V. Le, R. Monga, M. Devin, G. Corrado, K. Chen, M.A. Ranzato, J. Dean, and A.Y. Ng. Building high-level features using large scale unsupervised learning. *arXiv preprint arXiv:1112.6209*, 2011.

[100] Google maps floor plans. `https://maps.google.com/floorplans`. Accessed: 22/10/2012.

[101] A. Davids. Urban search and rescue robots: from tragedy to technology. *Intelligent Systems, IEEE*, 17(2):81 –83, march-april 2002.

[102] Shrihari Vasudevan and Roland Siegwart. Bayesian space conceptualization and place classification for semantic maps in mobile robotics. *Robotics and Autonomous Systems*, 56(6):522–537, June 2008.

[103] Frederic Bourgault, Alexei A. Makarenko, Stefan B. Williams, Ben Grocholsky, and Hugh F. Durrant-Whyte. Information based adaptive robotic exploration. In *in Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS*, pages 540–545, 2002.

[104] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *Proc. of Robotics: Science and Systems (RSS)*, Cambridge, MA, USA, 2005.

[105] A. Doucet, N. De Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 176–183. Morgan Kaufmann Publishers Inc., 2000.

[106] R. Sim. Stable exploration for bearings-only slam. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2411–2416. IEEE, 2005.

[107] Emily Whiting. Geometric, topological and semantic analysis of multi-building floor plan data, 2006.

[108] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, June 1998.

[109] P. Erdős and A. Rényi. On random graphs. I. *Publ. Math. Debrecen*, 6:290–297, 1959.

[110] R. Albert and A.L. Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[111] A.L. Barabási and Rékai Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[112] Xifeng Yan and Jiawei Han. gspan: Graph-based substructure pattern mining. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, ICDM '02, pages 721–, Washington, DC, USA, 2002. IEEE Computer Society.

[113] Gökhan H. Bakir, Thomas Hofmann, Bernhard Schölkopf, Alexander J. Smola, Ben Taskar, and S. V. N. Vishwanathan. *Predicting Structured Data (Neural Information Processing)*. The MIT Press, 2007.