Technical University of Denmark

Service Robots for Hospitals
Key Technical issues

Özkil, Ali Gürcan

Publication date:
2011

Document Version
Publisher's PDF, also known as Version of record

Service Robots For Hospitals: Key Technical Issues

PhD thesis 12.2011

DTU Management Engineering

Ali Gürcan Özkil
September 2011

DTU Management Engineering
Department of Management Engineering

Service Robots For Hospitals: Key Technical Issues

Ali Gürcan Özkil

PhD Thesis, April 2011

Summary

Hospitals are complex and dynamic organisms that are vital to the well-being of societies. Providing good quality healthcare is the ultimate goal of a hospital, and it is what most of us are only concerned with. A hospital, on the other hand, has to orchestrate a great deal of supplementary services to maintain the quality of healthcare provided.

This thesis and the Industrial PhD project aim to address logistics, which is the most resource demanding service in a hospital. The scale of the transportation tasks is huge and the material flow in a hospital is comparable to that of a factory.

We believe that these transportation tasks, to a great extent, can be and will be automated using mobile robots. This thesis consequently addresses the key technical issues of implementing service robots in hospitals.

In simple terms, a robotic system for automating hospital logistics has to be reliable, adaptable and scalable. Robots have to be semi-autonomous, and should reliably navigate in large and dynamic environments in the hospital. The complexity of the problem has to be manageable, and the solutions have to be flexible, so that the system can be applicable in real world settings.

This thesis summarizes the efforts to address these issues. Upon the analysis of the transportation tasks and how they are currently handled in hospitals, a navigation system is envisaged. Visual tags are a part of this system, and a survey was conducted to find out the most prominent ones to be used in mobile robot navigation. The concept of hybrid mapping is at the core of the solution, making it possible to efficiently represent the environment. Topological nodes greatly improve planning capabilities, and create a redundant layer for localization. The system features automatic annotation, which significantly reduces manual work and offer many advantages beyond robotics. Finally, this thesis outlines our contributions in representation of multi-floor buildings, which is a vital requirement to achieve robust and practical, real-world service robot applications.

Resumé

Hospitaler er komplekse og dynamiske organismer, der er afgørende for velfærdsniveauet i et samfund. Det ultimative mål for hospitaler er at levere sundhedspleje af høj kvalitet, hvilket også er det, der ligger de fleste af os mest på sinde. På den anden side er et hospital nødt til at orkestrere en række supplerende tjenester for at opretholde kvaliteten af sundhedsydelser.

Denne afhandling og dette erhvervs ph.d.-projekt har til formål at løse den mest ressourcekrævende logistik på et hospital. Omfanget af transportopgaver er enormt, og materialestrømmene på et hospital kan sammenlignes med en fabrik.

Vi mener at disse transportopgaver i stor udstrækning kan og vil blive automatiseret ved hjælp af mobile robotter. Denne afhandling behandler således de vigtigste tekniske aspekter ved at implementere servicerobotter på hospitaler.

Sagt på en enklere måde må et robotsystem til automatisering af hospitalslogistik være pålideligt, tilpasningsdygtig og skalerbart. Robotter er nødt til at være delvist selvstændige, og de bør pålideligt kunne navigere i store, dynamiske miljøer på et hospital. Kompleksiteten i opgaven bør kunne håndteres, og løsninger skal være fleksible, således at systemet kan anvendes i den virkelige verden.

Denne afhandling opsummerer bestræbelserne på at løse disse problemer. Efter analyse af transportopgaverne og hvordan disse i øjeblikket håndteres på hospitaler tages et navigationssystem i betragtning. Visuelle tags er en del af dette system, og en undersøgelse blev gennemført for at finde de bedste til anvendelse i mobil robotnavigation. Begrebet hybrid kortlægning er kernen i løsningen, fordi det gør det muligt effektivt at repræsentere miljøet. Topologiske knudepunkter forbedrer i høj grad planlægningskapaciteter og skaber et overskydende lag for lokalisering. Systemet har automatisk kommentering, som reducerer manuelt arbejde og tilbyder mange fordele ud over robotteknologi. Endelig beskriver denne afhandling vores bidrag til repræsentationen af bygninger med flere etager. Dette er en afgørende forudsætning for at opnå robuste og praktiske service-robotapplikationer, der er anvendelige i den virkelige verden.

Preface

This thesis has been prepared at Engineering Design and Product Development Section of DTU Management Engineering, the Technical University of Denmark, in partial fulfillment of the requirements for acquiring the Ph.D. degree in engineering. The project was funded by Force Technology and The Danish Agency for Science, Technology and Innovation.

The thesis deals with different technical aspects of service robots in hospitals. The main focus is on reliable, scalable and adaptable navigation methods for mobile service robots.

The thesis consists of a summary report and a collection of five research papers written during the period 2008–2011.

Kgs. Lyngby, April 2011

Ali Gurcan Ozkil

In memory of Torben Sørensen

Acknowledgements

This thesis has been made possible with the help of several people, whom I would like to thank for their efforts, ideas, support and participation; my advisors Zhun Fan, Steen Dawids, Jens Klæstrup Kristensen, Kim Hardam Christensen and Henrik Aanæs; Steen Ussing, Ulf Larsen, Ernst Christensen and Guri Breinholt from Force Technology, Jizhong Xiao from CCNY, all of the colleagues at the Section at DTU, helpful personnel of Bispebjerg Hospital, all the PhD students I shared offices, labs and lunches with, and Ingeborg Michelsen regarding her help with the resumé.

Table of Contents

Chapter 1

Introduction

1.1 Background

With an increasing average population age of societies, especially in Europe, a strain of the load in healthcare system is inevitable. In order to provide the expected healthcare service, several supportive services, such as logistics, have to be successfully coordinated to assure the quality and continuity of the healthcare. However, growing number of patients, increasing operational costs and shortages of qualified personnel makes it more and more difficult for hospitals to maintain the needed level of quality of the healthcare.

Studies conducted in the Bispebjerg University Hospital shows that the transportation system in the hospital is neither efficient nor sustainable [1]. Almost 30% of hospital resources are dedicated to logistics; making it the most manpower-demanding service in hospitals. A large volumes of goods; including pharmaceuticals, medical samples, food, mail and contaminated garbage, have to be distributed and collected at different units of hospitals on a daily basis. Due to a lack of personnel, even professionally educated nurses are involved in these tasks, adding burden to their already overloaded tasks.

This project aims to relieve the situation by analyzing the situation and proposing a system that can increasingly take over large parts of transportation tasks in hospitals so that the staff can be more focused on patient-related tasks. A tangible target of the project is to develop and validate a robotic transportation system to relieve the uncomplicated and manually performed transportation tasks.

1.2 Questions to be asked

Despite the vast amount of academic work, only a handful of mobile robots exist in the market. In addition to a few vanished products, the few available systems are simply adapted from industrial AGVs or only work in confined areas of hospitals [2-5]. It is therefore an important question to ask: What are the preventing factors for the wide application of mobile robots in the real world environment? The answer(s) to this have been investigated in the project.

Reliability is considered an essential issue of mobile service robots. Insufficient reliability undermines the use of mobile service robots [6]. Two other closely related issues: *adaptability* and *scalability* are also critical if mobile service robots reach commercial use.

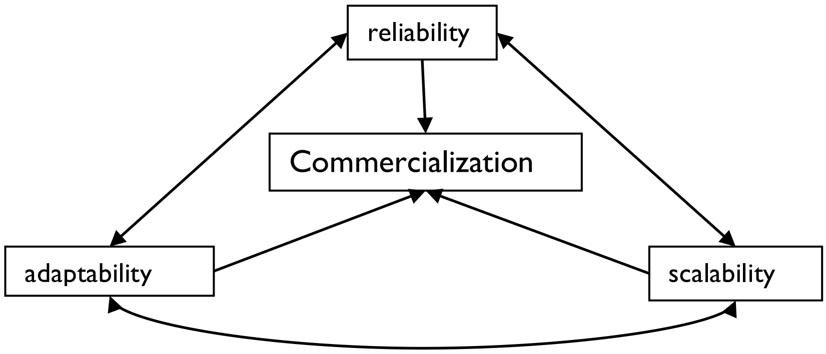

Figure 1, Three main issues for service robots

1.2.1 Reliability

Reliability is inversely related to complexity: One important way of achieving higher reliability is therefore to reduce complexity. In this project, a solution is envisaged based on a system view in which each large problem is divided into simpler tasks, which collectively accomplish higher levels of complexity. This solution actually stems from the well-known principle of 'divide and control', which is widely used in public transportation systems.

Therefore, it is essential that the complexity of the navigation task can be reduced to a level where simple solutions can be achieved. Other parts of the complexity, e.g. global planning should be allocated to the system management level in a flexible way.

Another important aspect of increasing reliability is to reformulate the software architecture for individual robots to make it more robust. Subsumption architecture [7] for example, is composed of behavior generating layers each connecting sensors to actuators. Unlike the traditional decomposition for an intelligent control system within AI, that breaks processing into a hierarchical chain of information processing modules from sensing to action, the subsumption architecture is more robust and *reliable,* as failure of one module will not necessarily cause the whole system to halt.

1.2.2 Adaptability

Daily environments in hospitals are not static and uniform. Application environment for mobile service robots will be different e.g. from one hospital to another hospital or from one department to another department in the same hospital. The environments are very dynamic and may include passing patients, nurses, doctors, or even other robots etc. It is therefore critical for the robots working there to be adaptable enough to achieve assigned tasks.

We refer to the dynamic and changing environment as unstructured environment [8]. It is important to define the level of unstructureness for a given environment. For example, we can find that the level of environmental unstructureness of the main entrance, which is crowded with patients and doctors, is quite different from that of an underground tunnel that is only used by the authorized hospital personnel. It can be assumed that different level of environmental unstructureness requires different level of adaptable intelligence from the robot.

Individual robots should *learn* the proper level of intelligence to fit their specific working environments, instead of being super-versatile to fit all types of environments, which has proved to be infeasible or undesirable by previous attempts [9], [10].

1.2.3 Scalability

One of the factors that usually draw the boundaries between research projects and real applications is scalability. Resources of any system are limited, and have to be allocated in harmony.

Hospitals are, to the most extent, located in large, multi-floor building complexes. Concerning service robots and their navigation, this brings substantial challenges in navigation. The magnitude of the data that has to be processed for mapping, path planning and localization in the dynamic and unstructured environment can be an important bottleneck for a robot, which can only have very limited computational power.

Considering all the above circumstances, 'divide-and-conquer' approach is vital to achieve useful service robots that will span considerable portion of a hospital. Consequently, the work presented in here addresses the issue of scalability in multiple dimensions.

This industrial PhD project aimed to provide guidelines encompassing three major components to resolve the critical technical issues – reliability, adaptability, and scalability, for design of hospital transportation system. The proof of concept is given by a functional prototype, and a number of research papers. It is expected that the concerted efforts of the partners and the results presented in this thesis will help improving the quality of service in future hospitals.

1.3 Organization of the thesis

The thesis is organized as a summary report accompanied with research papers that has been written during the course of the PhD project.

Robotic is at the core of this thesis, and it is a complicated field; therefore it is necessary to introduce some basics concepts that will pave the path to more detailed discussions. Chapter 2 aims to achieve this goal, by presenting robotics paradigms and fundamentals of mobile robot navigation.

Chapter 3 deepens the discussion on the robot navigation, providing the aspects of the software and hardware framework. This chapter also delves into the important technical characteristics of the methods that are further presented in papers.

Chapter 4 introduces the papers that are included at the end of the thesis. This chapter is not a mere summary; it also documents how the work presented in these papers has been extended with a unifying theme of reliable, adaptable and scalable service robots for hospitals.

Finally concluding remarks are presented in Chapter 5, and the thesis is finalized with research papers in Chapter 6.

Chapter 2

Basic Concepts

Robotics is an interdisciplinary field that borrows and repurposes many methods; which makes it often necessary to grasp a basic understanding of some of the key concepts. This chapter deals with explaining some of these concepts that will pave the path to the rest of the thesis.

2.1 Robotic Paradigms

A paradigm is a philosophy or set of assumptions or techniques, which characterize an approach to a class of problems. In this sense, the aim of a robotic paradigm is to organize the 'intelligence' of the system and control its actions.

A robotic system has three main set functions: SENSE, PLAN and ACT [11]. In simple terms; SENSE functions gather information from robot's sensors and produce a useful output for other functionalities, PLAN functions take these sorts of outputs or use robot's own knowledge to produce a set of tasks for the robot to perform, ACT functions produce actuator commands to carry out physical embodiment with the environment.

There currently are three mainstream paradigms in robotics, and they differ in the way they utilize three primitive functionalities. Figure 2 illustrates these paradigms.

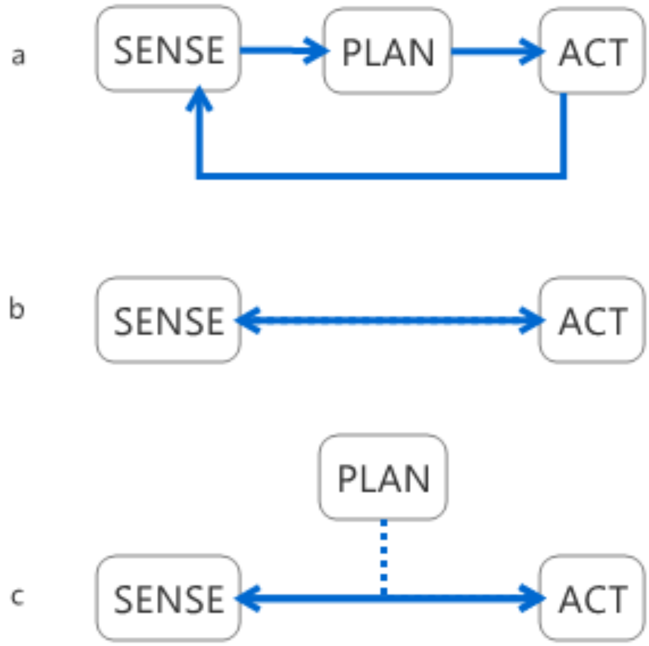

Figure 2, Three robotic paradigms; (a) hierarchical, (b) reactive, (c) hybrid

2.1.1 Hierarchical paradigm

The oldest method for organizing the intelligence is based on a hierarchical composition. Also called as the classical/traditional artificial intelligence, this method has been a dominating way to control robots through a human-like logical sequence of actions since the first mobile robots emerged [12].

Under this paradigm, the robot basically senses the world, plans its action, and then acts. Therefore, at each step it explicitly plans the next move. This model tends to construct a database to gather a global world model based on the data flow from the sensors, such that the planner can use this single representation to route the tasks to actions.

Two examples for the systems that used hierarchical control were the hospital robotic couriers Nestor [1] and Helpmate [13].

2.1.2 Reactive paradigm

Reactive paradigm came out as a reaction to the hierarchical paradigm in 80s. Hierarchical approach was based on an introspective view of how people think in a top-down manner. Reactive approach, on the other hand, utilized the findings of biology and cognitive physiology; which examined the living examples of intelligence [11].

In this approach, sensing is directly coupled to actuation, and planning does not take place. There are multiple instances of SENSE-ACT couplings, which can be also called as behaviors. The resulting action of the robot is the combination of its behaviors.

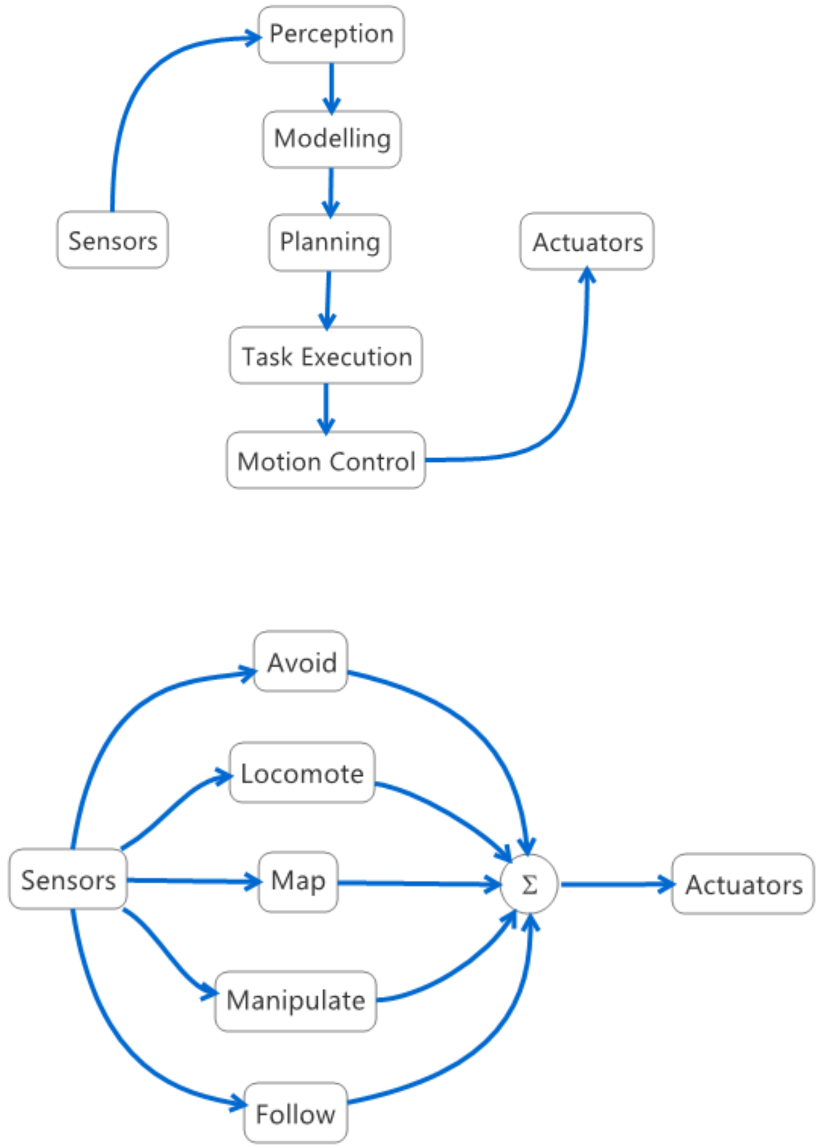

Figure 3, Hierarchical paradigm in detail (top), a reactive control paradigm example (bottom)

Brooks, in his seminal paper [7], described the main difference between these two approaches in the way they decompose the tasks. According to him, reactive systems decompose tasks in layers. They start with generating basic survival behaviors and then evolve new ones that either use the existing ones or create parallel tracks of more advanced ones. If anything happens to the advanced ones, the lower behavior will still operate, ensuring the survival of the system. This is similar to the functionality of human brain-stem that controls vital functions like breathing, which continues independently from high level cognitive functions of the brain (i.e. talking), or even in case of cognitive hibernation (i.e. sleeping)

Roomba vacuum cleaning robot [14] is the most obvious example that demonstrates the potential of reactive control.

2.1.3 Hybrid Paradigm

Hybrid approach was first exemplified by Arkin in 90s, in order to address the shortcomings of the reactive systems [15]. In this approach planning occurs concurrently with the sense-act couplings in such a way that tasks are decomposed to subtasks and behaviors are accordingly generated. Sensory information is routed to requesting behaviors, but it is also available to the planner for building a task oriented world model. Therefore, sensing is organized as a mixture of hierarchical and reactive styles; where planning is done at a concurrent step and sensing and acting are done together.

The hybridization brought up several architectural challenges, such as how to distinguish reaction and deliberation, how to organize deliberation, or how the overall behavior will emerge. Several architectures have been developed to tackle these issues, most of which mainly focused on behavioral management. It was found out that two primary ways of combining behaviors; subsumtion [7] and potential field summation [16] are rather limited, so other methods based on voting (DAMN) [17] , fuzzy logic (Saphira) [18] and filtering (SFX) [19] were introduced. The book 'Behavior Based Robotics [20]' is regarded as the most complete work on AI robotics, with a comprehensive list of such robot architectures explored in detail [11].

2.2 Autonomous Navigation

Autonomous mobile robot navigation can be characterized by three questions [21]:

- Where am I?
- Where am I going?
- How do I get there?

In order to tackle these questions, the robot has to:

- Handle a map of its environment
- Self localize itself in the environment
- Plan a path from its location to a desired location

Therefore the robot has to have a model of the environment, be able to perceive, estimate its relative state and finally plan and execute its movement.

An autonomous robot navigation system has traditionally been hierarchical, and it consists of a dynamical control loop with four main elements: Perception, Mapping/localization, Cognition and Motor Control (Figure 4).

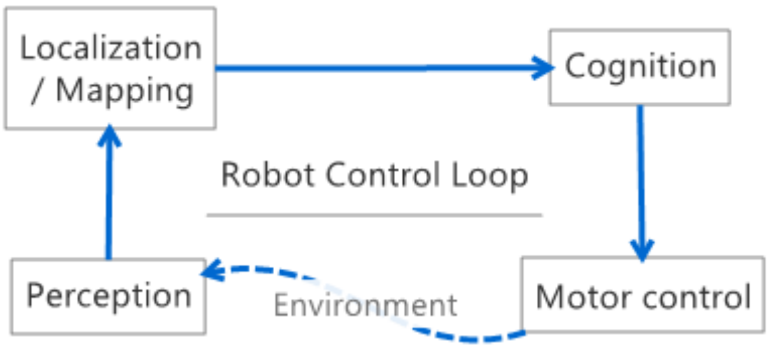

Figure 4, Autonomous navigation problem

This section aims to summarize these elements and give an overview of relevant problems to be addressed.

2.2.1 Perception

First action in the control loop is perception of the self and the environment, which is done through sensors. Proprioceptive sensors capture information about the self-state of the robot, whereas exoprioceptive sensors capture information about the environment. Types of sensors being used on mobile robots shows a big variety [22], [23]. The most relevant ones to service robots can be briefly listed as: encoders, gyroscopes, accelerometers, sonars, laser range finders, beacon based sensors and vision sensors.

In theory, navigation can be realized using only proprioceptive sensors, using odometry. It is basically calculating the robot position based on the rotation of wheels and/or calculating orientations using gyroscopes/accelerometers. But in real world settings, odometry or inertial measuring systems perform poorly due to unbounded growth of integration errors caused by uncertainties.

It is also possible to navigate using only exoprioceptive sensors. One such realization of this approach is the Global Positioning System (GPS); which is being successfully used in vehicle navigation systems. The problem with GPS and its upcoming, European counterpart Galileo [24] is that these systems require a direct line of sight to the satellites on earth orbit. Therefore these systems are especially inapplicable to indoor applications.

Shortcomings of GPS system led researchers to several ground based approaches. Several alternatives have been developed based on e.g.: Radio beacons[25], Wireless Ethernet[26], GSM networks [27], Wireless Sensor Networks (WSN)[28] , RFID tags [29], or laser reflectors [30]. Such methods can ease the problem of navigation, but they might require substantial amount of environmental modification. This makes them inflexible and costly to install and maintain. Due to such reasons, many researches focused on solving the robot navigation problem in unmodified environments.

Many of the state of the art techniques for navigation in indoor environments uses combinations of proprioceptive and exoprioceptive sensors and fuse them using probabilistic techniques.

2.2.2 Environmental Representation

How the environment is represented is one of the most important factors in navigation. Depending on the characteristics of available sensors and data acquisition system (such as range and resolution, update speed, bandwidth) several methods exist to store and handle 'world models'.

Simplest way of representing an environment is using raw sensor data. Information coming from sensors i.e. laser range scans, are sequentially stored in the same type of data they are acquired. In such a case, the problem is the low distinctness of the data. Many of the objects will be sensed and represented similarly, without distinguishing their 'features'. Also, this approach eventually results in a large volume of data with time, which brings up computational challenges.

Alternatively, features can be used for modeling. In simple terms, a feature is a distinguishable characteristic of an object or the environments, such as corner of a wall or color of a door.

Complexity level of features is an important factor for navigational purposes. Using low level features such as lines/circles will generate a smaller database compared to the previous approach, yet with a moderate amount of ambiguity associated. More complex features in the forms of i.e. patterns/objects can even decrease the size of the database with lesser ambiguities. Too much complexity of features can also have adverse effects. First, it might yield difficulties in detection and require high computational resources. And secondly, it will result in very small databases, which might not entirely capture the characteristics of the environment.

Range sensors have been the dominating choice for environmental sensing on robots. Early works extensively used sonar arrays for distance sensing, but the limitations with range and resolution of sonars severely affected functions of mapping and localization. Time-of-flight laser scanners later became widely applicable to mobile robotics, but their scanning field is restricted to a horizontal plane, which in turn yields to poor world representation [31]. This limitation was tackled by using oscillating the laser scanners [32][33][34] or multiple lasers with complementary placements [35] to achieve higher dimensionality in range sensing. Yet, these systems are rather expensive and complex to utilize in a real world robotic application. Finally, different vision based approaches has been emerged in the last decade to extract metric information from the environment using imaging sensors. Stereo systems have been long investigated for 3D range sensing, whereas most of recent work is based on monocular systems that can extract metric information from the optical flow detected by the camera, and multiple view geometry.

2.2.3 Maps Used in Mobile Robot Navigation

Idea of using maps for mobile robot navigation has been existed for quite some time, and roboticists have developed several types of maps for different needs based on how they can represent the environment. Buschka [36] classifies existing map types as follows:

- ***Metric Maps****:* Maps that carry distance information that corresponds to actual distances in the environment. Such a map can give a distance of a path or size of an object.
- ***Topological Maps***: Maps where the environment is modeled according to its structure and connectivity, and often represented as a connectivity graph.
- ***Sensor Level Maps****:* Maps that are derived directly from the interpretation of the sensor inputs from the current position.
- ***Appearance Based Maps****:* Maps that functionally describe a position from sensor data.
- ***Semantic Maps****:* Maps, which are oriented for high-level decision-making, and contain information about objects and their relationships with the environment.
- ***Hybrid Maps****:* A combination of different types of maps. Hybrid maps also need to glue elements that represent the same part of the environment in combined maps.

Following section elaborate on metric, topological and hybrid maps, with are the most commonly used types in mobile robotics.

2.2.3.1 Metric Maps:

From the control perspective, metric maps are useful when metric accuracy is necessary for i.e. precise localization or optimal path planning. Depending on the environmental representation, a metric map can be either feature based or grid based [37].

A metric feature based map is basically built upon the features that can be reliably observed in the environment. In [38-40] typical features of indoor environments such as walls, edges or corners are used for mapping of indoor environments.

In a grid based metric map, environment is divided in to a matrix of sub cells, where each cell represents a portion of the environment. A cell regarded as occupied if an object exists in the corresponding area in the environment. Moravec and Elfes developed a common way of representing occupancy is using probabilities [41]. Saffiotti used fuzzy sets where occupancy and emptiness values are held separately [42]. In [43], grid is represented using histograms where each cell holds a value of how often a sensor has detected it. Stachniss and Burgard developed a coverage map, where each cell holds a value representing how much it is covered by an obstacle [44].

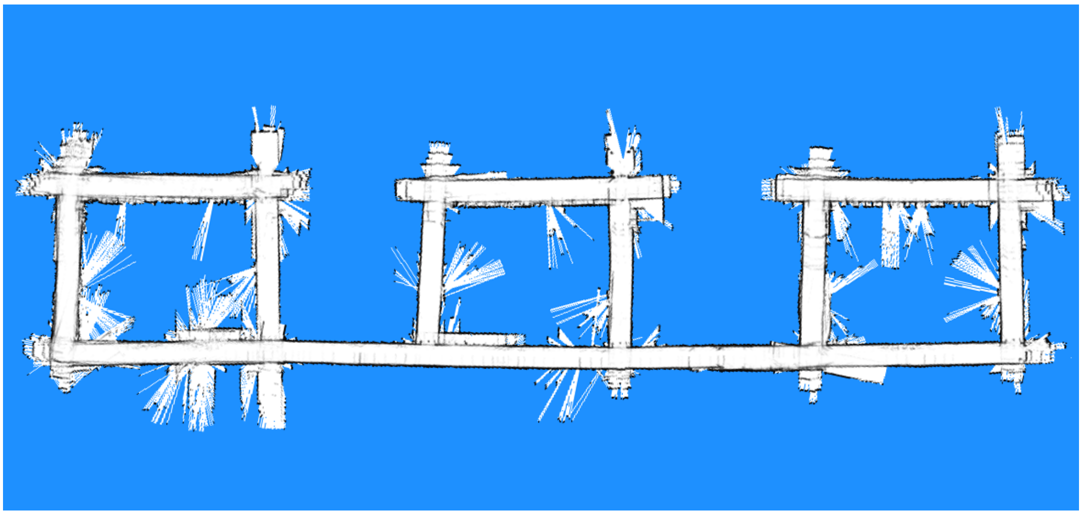

Figure 5, Metric grid map of DTU 402-404

2.2.3.2 Topological Maps:

Topological maps describe how places in the world are connected or related to each other, thus represents the structure of the environment. Two elements that constitute a map are nodes; which represent the places, and edges; which represent connectivity. In practice, most topological maps are also augmented with some metric information on its nodes. Due to its simplicity in construction, topological maps are better suited for problems that require searching. (See Figure 6 for an example)

Apart from robots, humans might also need to interact with the topological maps for robot navigation. Different characteristics of the environment have been used by researches such as rooms or corridors as nodes and doors or passageways as edges. Thrun [45] preferred to use places with 'significant features' as nodes. Fabrizi [46] defined a node as a 'large open space'. Duckett [47] proposed a system where a new node is placed after robot has travelled far away from the previous one.

Topological maps, such as reactive control paradigm, were inspired by biological studies of insects and animals. It can be also claimed that a topological map will be the best suitable for a behavior based navigation system.

Figure 6, A topological map: S-tog network in Copenhagen

2.2.3.3 Hybrid maps:

Since metric and topological maps are of fundamentally different types, both have advantages over each other. Table I summarizes this comparison.

It is clear to see that what is an advantage for one approach is a disadvantage to the other, which constituted the motivation to develop hybrid maps. The idea came to the scene as early as 1978 [48], but it has only been a decade that such maps emerged in an increasing number.

Table 1, Comparison of Metric and Topological maps [36]

	Metric Maps	Topological Maps
Pros	• High accuracy for localization and path planning • Possibility to make optimal routes • Easy to build, represent and maintain for small environments • Layout is easily readable for humans	• Easy to scale up for large environments • Very suitable for planning • Sensor precision and reliability is not as important • No need for precise position estimation for map building • Good interface to symbolic problem solvers
Cons	• Difficult to scale up for large environments • Costly path planning • Need for reliable sensors • Need for precise position estimate for map building.	• Low accuracy • Possible suboptimal paths • Difficult to build and maintain

Two types of hybrid maps are parallel maps and patchwork maps. A parallel map constitutes of at least two different maps that represent the same area in an environment. Most parallel maps are constructed automatically by extracting a topological map from a metric one. Thrun utilized Voronoi diagrams in the empty parts of a grid map in [45]. A similar approach was carried in [46] by using image processing. The opposite approach, extracting metric maps from topological nodes had also been presented [49]. An interesting multi-layered hierarchical parallel map representation is developed in [50], [51] where the main focus was efficient localization. The map is called 'Annotated Hierarchical graph' and it consists of hierarchically ordered topological maps, supported with local metric patches in the lowest layer. Nieto also developed a novel kind of parallel map, which consisted of an augmentation of a feature based metric map and a grid based metric map [52]. While the first were used for localization, the latter was used for optimal route planning.

A patchwork map is a representation, where the environment is globally symbolized by a topological map and a set of metric map patches. This kind of maps can be easily scaled up, thus representing really large environments; yet perform fine metric localization due to the patches.

Several patchwork maps simply connect small sized metric maps based on topology [53]. Thus nodes do not correspond to any particular environmental structure. More elaborate patchworks used openings between i.e. rooms and corridors as the node features [54]. In [55], [56] similar approach is used for node selection, and the rest of the topology is completed using 'Reduced Generalized Voronoi Graph'. Aguirre developed a complex patchwork map in [57], where two kinds of metric maps where used in each room, which acted as nodes in the topology.

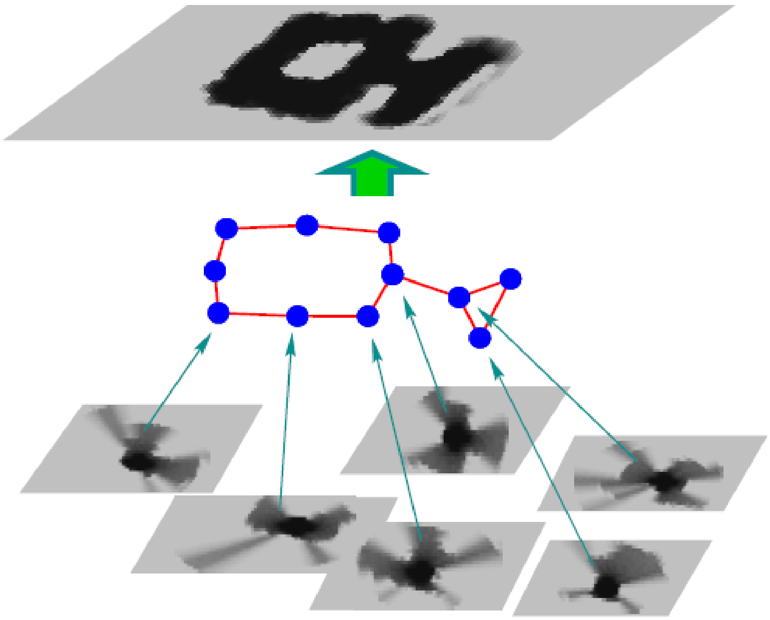

Figure 7, A hybrid map: Unified metric map is extracted from topological nodes [49]

2.3 Mapping and Localization

2.3.1 Mapping

A mobile robot requires a representation of the environment for autonomous navigation in the form of a map. Based on the environment characteristics and the type of the map, it is possible to build robot maps using existing maps by other means. But in most of the cases, the robot needs to build a map of the environment in a subsequent training phase.

Metric grid maps are the most commonly used types of maps in mobile robot navigation. Building metric maps requires estimating the initial position of the robot, and updating the cells of the map as the new sensory information is acquired. The most trivial approach is to use odometry for position estimation. As explained previously, estimation error accumulates by time in odometrical systems. The apparent idea [58] to address this problem is to use the map, which is being built at that moment, for correcting the estimation, which is now coined as *Simultaneous Localization and Mapping* – SLAM.

Particularly difficult part of the SLAM problem with the grid maps is that the cell positions in a grid map are static. Therefore, if a robot recognizes a place it has already been during mapping (loop closure), it might see that its position is off and needs to be corrected. On the other hand, to correct the grid map, the entire map should be traced back and recalculated based on the new information.

An evident method is to build the map sequentially by first localizing and then rebuilding the map based on adjusted positions [59]. Genetic algorithms are also used for mapping. Duckett developed a method [60] where several maps are generated with slightly altering paths, and then a genetic algorithm is used to select the best maps and combine new paths to test. In [61], a new grid map representation is generated where the cells are able to hold multiple hypotheses about the map. The least probable hypotheses are later removed in a map update stage. Rao-Blackwellized particle filters, introduced in [62] became a popular choice for building grid maps. In this approach, a number of maps based on single particles are being carried and updated simultaneously. Recent improvements on this method permitted to reduce the number of particles to still get good results [63], [64]

Feature maps differ from grid maps in the sense that sensor data is used to extract features before the mapping stage. These features are then compared to the ones in the map so that either the new feature is added to the map or the existing features in the map are updated accordingly, or used for correcting the position estimation. Many of the solutions are based on the approach presented in [65]. The most significant developments around this method are based on how the Kalman filter is utilized for position update. Information filter is introduced to ease the computation burden in [66], [67]. Also unscented filter is used in [68], [69] to cope with the nonlinearities.

Building topological maps can be done in two different ways; by using sensor data or by using another type of map. Choset used a generalized Voronoi graph as a map in [55]. The map is constructed my moving the robot in the environment to construct the nodes of the map, and visited nodes are detected by matching their "signatures" to the previously acquired ones. Thrun et.al used the latter technique in [56], where they preprocessed the grip maps to threshold the occupancy probabilities to further generate Voronoi diagrams on the empty areas of the map. Local minima found in the Voronoi graph are used to partition the grid map into nodes.

2.3.2 Localization

Localization is the task of finding the position of a mobile robot in an environment, based on its representation.

2.3.2.1 Metric localization

Localization can be defined as the task of estimating the robot's pose in the world, given a-priori a map. The estimate, or *belief*, is often augmented with some measure of uncertainty that can arise from several factors, such as wheel slippage or uneven floors. The belief is updated when the robot performs an action or makes an observation. A robot action (i.e. movement) increases the uncertainty (due to integration errors), whereas observations often reduce the uncertainty of the robot pose.

Localization is tightly coupled to how belief is represented and estimated. Most of the robotic systems use planar maps. The main reason for that is to decrease the complexity of the problem by reducing dimensionality of the robot pose vector from 6-D (x, y, z, pitch, roll, yaw) to 3-D (x, y, yaw).

One approach to solve the problem is position tracking, where the belief of the robot is reduced to a single pose. The position is estimated in a single hypothesis, and whenever an action or observation occurs, the hypothesis is updated. Therefore, the initial position of the robot must be known to be able to track the position. Kalman filter [70], and its variants are widely used in position tracking. In [71] sonars range finders are used for line extraction and a Kalman filter is used for matching. In [72], an extended Kalman Filter is used to match raw sensor data with a feature based metric map. Fuzzy logic is also used for representing uncertainty in position tracking in [73].

Position tracking problem deals with a single pose, therefore representation is simple, and update calculations are computationally cheap. But this technique requires that the initial position is known. In addition, if the measurements become vague, the position can be lost.

The alternative solution to position tracking is to represent multiple hypotheses of the pose, which is often called as global position estimation. In this approach, the initial position is not needed to be known, but due to the high degrees of uncertainties imposed by multiple hypotheses about the pose of the robot, position estimation is computationally expensive.

In [74] several position candidates are tracked using Kalman filters. The number of hypotheses adapts the uncertainty of the localization. Safiotti et. al. used fuzzy sets to represent uncertainties to carry out multi-hypothesis tracking in [75]. In [76] Markov localization is introduced, where each cell of a grid map holds a belief of how much the actual position of the robot is in that cell. In this approach, the localization grid map represents a probably density function (PDF) of the belief of localization. As the robot moves or observes, cells are updated using Bayesian updating. In [77], the method is further modified to overcome the heavy computational cost of the approach. An alternative is proposed in [78], where the updating is based on fuzzy logic instead of Bayesian inference. In [79] PDF used in localization belief is represented as a set of samples. This approach reduced the number of calculations compared to Markov localization, while it is still possible to perform global localization. This approach is called Monte Carlo localization, and further improved in [80-82] to decrease the number of samples needed.

2.3.2.2 Topological Localization

A topological map is consisted of nodes and edges. Therefore, topological localization is the task of finding in which edge or node the robot is. Apart from the environmental representation (i.e. how the edged and nodes are defined), topological localization requires reliable place recognition and detection of edge traversal. In [83], nodes are defined based on the sudden changes in the behavior pattern of the robot. For instance, if the robot is following a wall, and after a while it encounters an obstacle so it as to perform another action, that particular place is defined as nodes. Nodes are identified using features such as distance travelled since the last node and the 'signature' of the node given by sonar sensors. In [84], the nodes are recognized using a similar signature approach, and then these signature are learned using a growing neural network. Localization is then performed using signature matching and odometry. In [85], a local topological map is built and then compared to a global one, to obtain the most likely position. In [86], an omnidirectional camera is used for performing topological localization. Queried images are compared to images stored in the map. Image histograms are used as global features for image representation, thus the amount of information stored is highly reduced.

2.4 Cognition and Path planning

Robot cognition very much depends on the general use of the robot. In the context of this report, problem setting is defined as autonomous mobile robot navigation. Therefore, the discussion is focused on path planning for navigation.

Path planning can be defined as searching a suitable path in a map from one place to another. Depending on the map type, it is possible to follow different strategies for planning paths.

Metric maps are useful for planning precise paths. Due to metric information associated, it is possible to find nearly optimal paths using metric maps [36]. There exists several different methods for path planning, but they are based on a few general approaches. Latombe classifies these approaches in [87]as follows:

Road map: a road map is a collision free set of path between a starting position and an ending position. Therefore, they describe the connectivity of robot free space on the map. One method to construct road map is based on visibility graphs [88].

In this method, path is incremented from one point to other points that are visible from the first point. Another method is to construct a Voronoi graph, which tries to maximize the clearances between the robot and obstacles [89].

Cell decomposition: free space in the map is divided into non-overlapping cells, and a connectivity graph describes how the cells are connected to each other. The result is a chain of cells, which also describes the path. Therefore, formation of cells plays an important role in planning the path. In [87], trapezoidal decomposition is used, where a polygonal map is divided into trapezoidal cells by generating vertical line segments at each corner of every polygon. In [90] qualitative spatial reasoning is used for path planning, which is inspired of the way humans find their paths with imprecise knowledge. Cell decomposition is also a suitable method for area coverage, where the planner breaks down the target area into cells to be all traversed. Applications of this approach can be listed as i.e. lawn moving, snow removal or floor cleaning [91]

Potential field: A potential field function is defined an applied over the free space on the map, where the goal acts as an attractive potential (sink) and the obstacles act as repulsive potentials (sources). The path is then derived based on the derivative of the potential field, where the steepest direction is followed. This approach was first developed for online collision avoidance in [92]. It is combined with a graph search technique in [87] for path planning.

Topological maps are well suited for planning paths. Graph search algorithms, such as A* [93] or D*[94], can be used the plan the shortest path on a topological map. In most of the cases, the number of edges and nodes are moderate, so the path planning can be performed very quickly. Path finding time is even further shortened in [95] by preprocessing all paths and storing them in a lookup table. In [85] wave-front algorithm is used for both path planning and collision avoidance. In [96], [97], planning on very large maps is described in the context of hierarchical topological maps.

2.5 Motion Control

Motion control is the final phase in the robot control loop, where the high level plans generated in the previous phase are translated into robot movements. Therefore, this level processes abstract motion commands and produces low-level commands for controlling motor speeds.

Obstacle avoidance is of particular interest, and it can be classified under motion control. It is one of the key issues to successful mobile robot applications, as it ensures the safety of both robot and surrounding entities. Obstacle avoidance strategies range from primitive algorithms that just stop the robot when an obstacle is detected; to complex ones that enable robot detour the obstacles.

Borenstein introduced vector field histogram (vfh) algorithms fro obstacle avoidance tasks in [43], based on local potential fields. In this approach, first the range data is continuously sampled, and a two dimensional local grid is generated to represent the environment. In the next stage, one-dimensional polar histogram is extracted from the local grid in terms of angular sectors with particular widths. Finally, this one-dimensional histogram is threshold and the angular sector with the highest density is selected as the direction. Sped of the robot is also adjusted in correlation with the distance from the obstacle. In [98], the algorithm is improved by incorporating the kinematics of the robot as the original algorithm assumes that the robot is able to change its direction instantaneously (named as vfh+). The algorithm is further improved and coined as vfh* in [99]. In contrast to vfh and vfh+, which are purely local algorithms based on current sensor readings, vfh* incorporated A* graph search algorithm to consider more than immediate surroundings of the robot.

In [100], dynamic window approach is introduced as an obstacle avoidance method. Kinematic constraints of a Synchro drive robot are taken into account by directly searching the velocity space of the robot. The search space is further reduced to a dynamic window, which contains those velocities that can be achieved by the robot, given its velocity and acceleration. Finally, this window is searched for a velocity, which aligns with the target direction of the robot. In [101], the method is adapted to holonomic robots, which allowed high-speed obstacle avoidance with high maneuverability.

Finally, nearness diagram is introduced in [102], which is based on heuristic rules that are inferred from possible high and low safety situations that the robot can end up. Based on five rules (two low and three high safety situations), five behaviors are defined, where robot compares its current situation to these predefined ones and executes the appropriate behavior. It is shown in [103] that this reactive approach can perform well in cluttered environments with narrow passages, as compared to previous approaches.

Chapter 3

Navigation Architecture

This chapter deals with the framework that is used for navigation. The framework utilizes the strengths of two most commonly used exteroceptive sensors in robotics [22]: Laser range finders and cameras. The framework also combines navigation with metric maps based on laser range finder readings and navigation with topological maps based on "places of interests" that are extracted from captured images.

3.1 Robot Navigation - Using Metric Maps and Laser Range Finders

A robot is a complex mechatronic system, which is composed of many hardware and software components that needs to be orchestrated in (near) real time. Therefore, a software architecture is needed to synchronize simultaneously running processes and abstract hardware components. Luckily, there exist a few free or open source projects that fully or partially address these issues. Consequently, the practical implementation and the software structure of the software architecture are sketched in here; since it is the best way to describe the navigation framework.

3.2 The Software Architecture

Among other alternatives such as Microsoft Robotics Studio (MSRS) [104], Player/Stage [105] and Mobile Robot Programming Toolkit (MRPT) [106]; Carnegie Mellon Navigation Toolkit (CARMEN) [107] had been selected as the platform to improve upon at the beginning of the project. The main reason for selection was because CARMEN mainly focuses on the navigation problem instead of providing a broad and generic solution, and it is natively supported by the experimental hardware that was selected for the project (Pioneer 3-DX robots from ActivMedia [108]).

3.2.1 Modular Framework

CARMEN software platform is targeted for Linux operating systems, written in C programming language. It is heavily based on a client-server architecture, where the overall robot control task is decomposed into several smaller tasks that are handled by individual clients. In addition to a few other advantages of this approach, it mainly facilitates a distributed system and it permits failsafe operation where only the related processes are affected in case of the failure of a component.

A central process is responsible from inter-process communication (IPC), where clients publish and request information through. Under the hood, IPC takes care of opening and closing TCP/IP sockets, registering, sending and receiving packets, serializing/de-serializing the data; and therefore provides a high-level support for creating client/server systems. Figure 8 shows the operating principles of IPC.

Modules of the CARMEN can be grouped in four categories: Essential processes, primary processes, navigational processes and complementary processes. Figure 9 and Figure 10 shows the relationships between these processes.

3.2.1.1 Essential Processes

The main process in the toolkit is the "*central*", an Inter-Process Communication module, where all sorts of information that are produced by sensors, actuators and software components pass through.

Along with the *central*, the *parameter server* is the second process that needs to be running all the time for the other processes to function. *Parameter server* is responsible from providing parameters that are needed by several other processes. These parameters are numerous, and they vary from hardware settings such as the communication ports to be opened and the speed of communication, to operational settings such as maximum permissible speed of the robot or the number of particles to be used in the localization process. All of the parameters are stored in a single text file and this file is passed as an argument to the *parameter server* process at runtime.

3.2.1.2 Primary Processes

On top of the essential processes there are a number of primary processes that needs to be run for the actual functionality of the system.

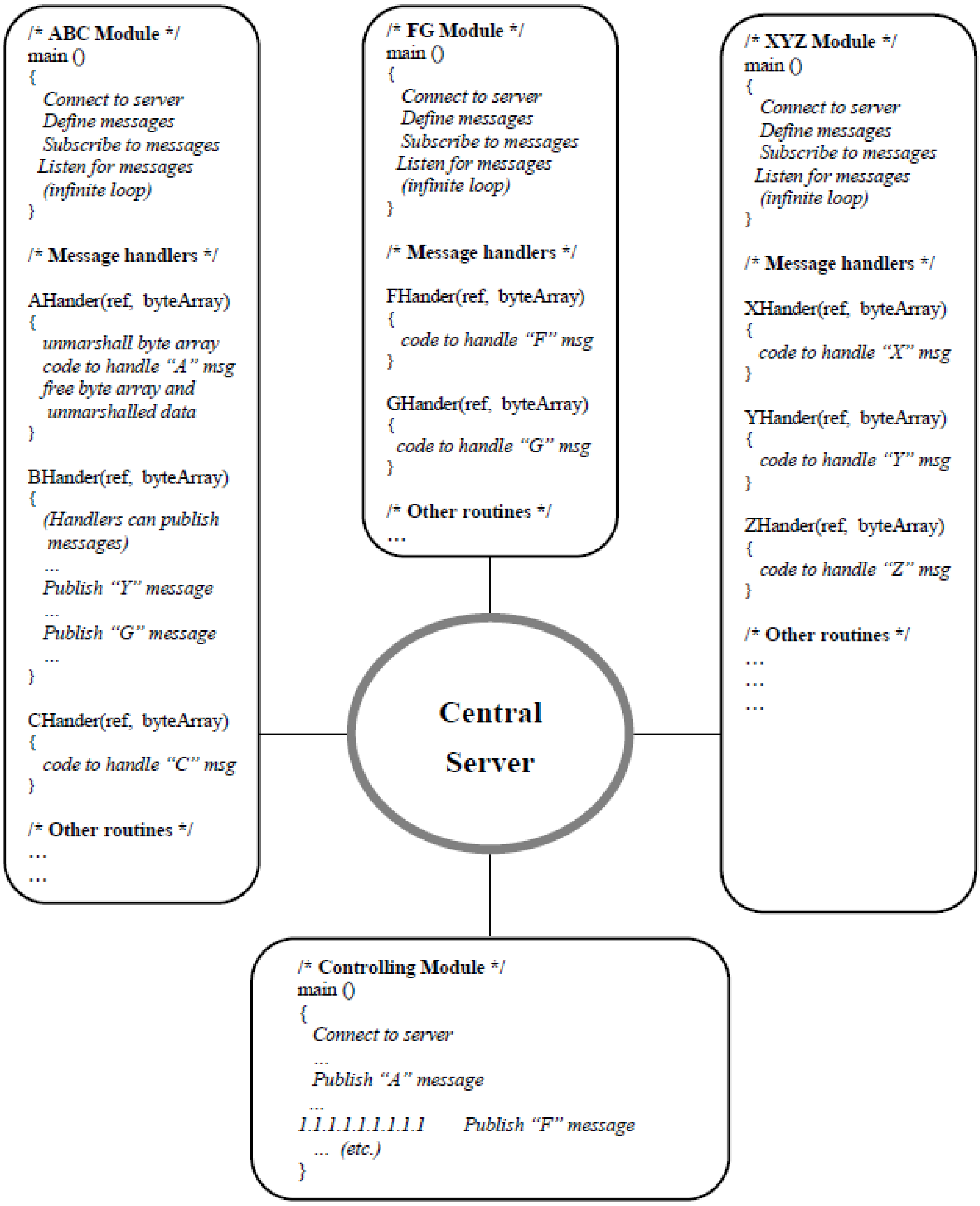

Figure 8, Inter-Process Communication architecture

The *robot* process is responsible from providing the core functionality. It abstracts the hardware of the robot by providing an interface to a generic robot object. It also provides a collision avoidance system, which stops the robot in front of the obstacles.

The *pioneer* process belongs to a family of processes that handles the actual communication between actual robot base and the *robot* process. It translates the high level motion commands to low level motor actuation commands, and serves proprioceptive information (such as odometry readings, battery state etc.) to the *robot* process. There are other similar processes, which are desgined in the same manner to interface with other hardware platforms.

The *laser* process interfaces a real laser sensor to the framework and publishes range readings that come from laser(s). A mobile robot can utilize multiple lasers, which can be of different types (SICK LMS or Hokuyo URG series). Positions, orientations and other properties of the laser(s) are specified in the parameter file.

These three processes are the primary processes that make it possible to command a physical robot through CARMEN platform. At this point, the *robot* process can be used to pass motion commands to the robot by direct user input through i.e. computer keyboard.

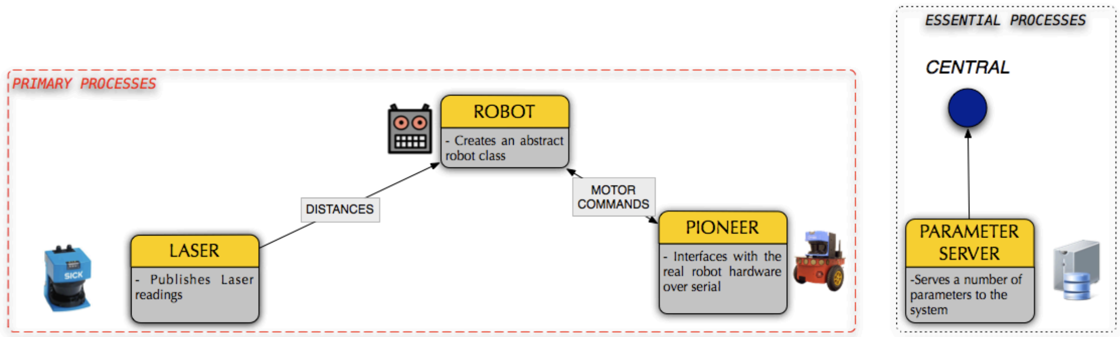

Figure 9, Essential and Primary Processes

3.2.1.3 Navigational Processes

The navigation problem can be solved by answering three repeatedly stated problems: *(i) Where am i? (ii) Where am I going? and (iii) How do I get there?* To achieve (semi) autonomous navigation, a few more processes are needed to answer these questions.

Robot navigation in the context of CARMEN is map-based navigation, where maps represent the environment in forms of metric occupancy grid maps. The *map*_server process does exactly what it is named for: serving an occupancy grid map to the requesting system processes.

Similarly, the *localize* process estimates the pose of the robot in the environment. It requires a map published by the *map* process, range readings published by the *laser* process and the motion commands (odometry) published by the *robot* process. The *localize* process implements a classical particle filter scheme to estimate the pose (the position and the orientation) of the robot. As a result, this process provides the estimated pose of the robot and the associated uncertainties to the estimation.

Finally, the *navigator* process takes care of the rest of the job by planning a path to the robot from its current position to a (reachable) destination (on the map). It requests map and pose information and estimates a number of waypoints for the robot to follow. These waypoints are passed to the *robot* process, which are translated to velocity vectors and further passed to the robot hardware.

The basic problem with autonomous robot navigation in real environments is the dynamic changes in the environment, which cannot be reflected to maps. Therefore the *navigator* process needs to additionally incorporate range sensor information and frequently re-plan the path based on the changing information.

3.2.1.4 Complementary Processes

The above-mentioned processes are the necessary and sufficient processes to achieve a semi-autonomously navigating robot. But CARMEN provides a few extra modules that complement the navigation problem.

The most basic of these processes is the *logger* process. What it simply does is to asynchronously record all the published information in a log file. The messages from different processes are joined, time-stamped and sequentially listed by the *logger*.

The *vasco* process (named after the famous cartographer *Vasco De Gama*) is the most distinguished of the complementary processes in the CARMEN package as it can be used to process the recorded log files to generate occupancy grid maps. It utilizes a simple scan-matching algorithm to estimate the positions of the walls and other static objects in the environment and correct the pose of the robot. That being said, hands on experience with the *vasco* showed that it is only useful in a very limited scale. Therefore in this work, an additional library called *gmapping* is adopted for this task CARMEN. It is based on an "*Improved Rao-Blackwellized Particle Filter*" which is described in detail in [17]. The *gmapping* process works similar to *vasco*; it processes a log file and generates an occupancy grid map.

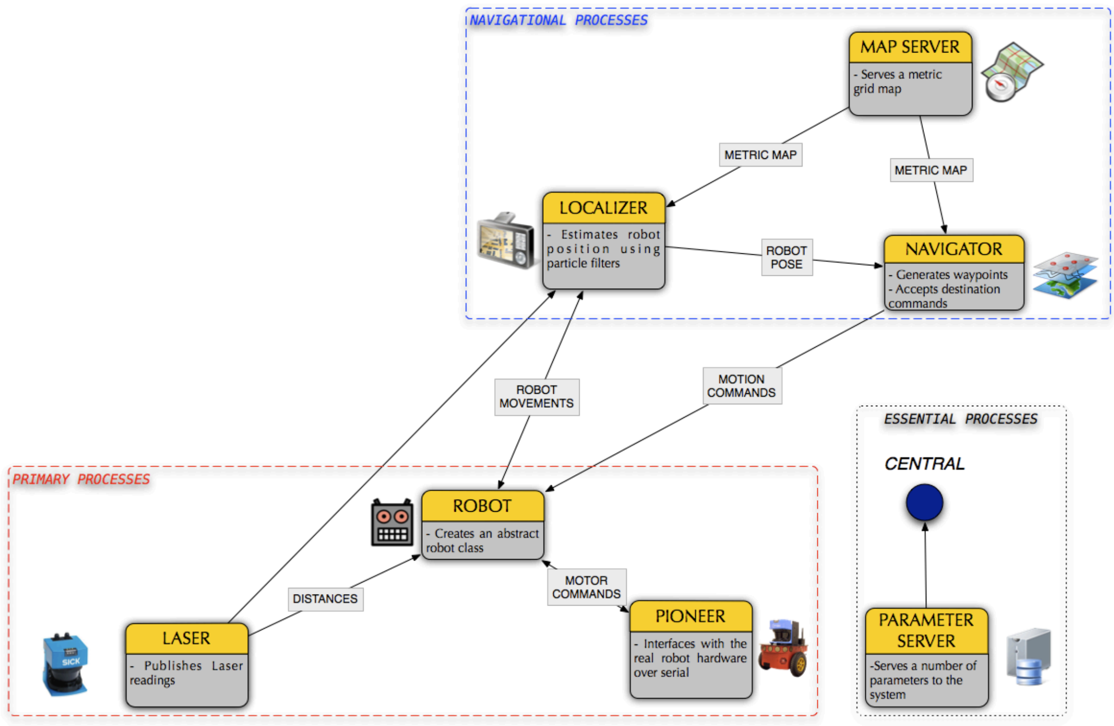

Figure 10, Navigational Processes

The *simulator* module facilitates a simulation environment to the rest of the system. Instead of using the specific robot process, it can be used to simulate the robot hardware and therefore smoothly run the rest of the processes without any need for alteration.

The rest of the complementary processes can be briefly listed as:

a. *camera:* publishes images
b. *joystick:* interfaces with a gamepad/joystick for manual navigation
c. *log_playback*: playbacks recorded log files as they are recorded at the time of playback.

3.3 Robot Navigation with the Help of a Camera

Cameras have been extensively used in robotics for the purpose of navigation. There are numerous methods that use cameras for extracting metric information from the environment [109] or to recognize places for topological navigation [31]. To improve the outlined navigation framework and achieve a useful system for service robot applications; a supplementary vision based topological navigation approach is proposed.

The underlying idea is very simple: Creating a hybrid map by augmenting metric information with topological nodes. Such a map will help managing the complexity of the overall environment, and help the robot to correct its position estimation whenever a node is encountered.

Due to a number of reasons that are further stated, visual tags are chosen to be used in accordance with the topological nodes. Despite the need for an initial installment in the environment, visual tags offer a number of advantages. First, they very easy to distinguish and recognize; which makes them robust and reliable. There is no need for initial training and testing, making them easily deployable. They are very easy to install (a printed paper and a piece of tape) and they require much less computational power compared to methods that utilize 'natural landmarks'.

The decision of using artificial marks for topological navigation comes with another question: What kind of artificial mark to use? This question is addressed in the paper: "*Visual Tags and Mobile Robots: A Review*", where several types of artificial marks are investigated. The concluding remark of the paper can be stated as follows: It is best to use augmented reality tags since they are designed to detect markers in a similar setup to a mobile robot; but 2d-barcodes also have a great potential due to their possibility of storing any type of data.

Consequently, it is decided to develop the first iteration of the navigation system with the markers and marker detection algorithm from the augmented reality project ARToolKitPlus [110].

The system consists of four modules: Node Map Maker, Tag-Identifier, Place Recognizer and Re-Localizer. These modules are described as follows.

3.3.1 Node-Map Maker (Topological map generation component)

This component is used for processing metric maps and combining them together with user supplied topological node data, i.e. marker IDs. The input of the component is the generated metric map and user supplied topology information. The output of the component is a hybrid metric-node map.

3.3.2 Tag Identification component

This component is used for detecting the existing markers on the supplied video stream. Multiple instances of this component can run and marker detection with multiple cameras is possible. The input is a video stream (or a sequence of images), and the output of is the ID numbers and poses of the markers on the input image.

3.3.3 Place Recognizer (Topological localization component)

This component is used for estimating the position of the robot using the hybrid map and the tag identification component. It simply reveals the relation between identified marker(s) and the relevant information stored in the database (topological map). The inputs of this sub-system are the ID of the marker detected and the topological map and the output is the position of the marker relative to the global map.

3.3.4 Re-Localizer (Integration component)

This component fuses the information given by the topological localization component and the particle-filter based metric localization process, which yield to a metric re-localization routine under the presence of an identified marker. The inputs are the estimated position of the robot and the estimated position of the identified marker. The output of the component is corrected position of the robot.

Using the above-mentioned software components the main navigation framework is achieved. The resulting system is graphically presented in Figure 11; and it constitutes the base for the further methods that are described in the following chapters

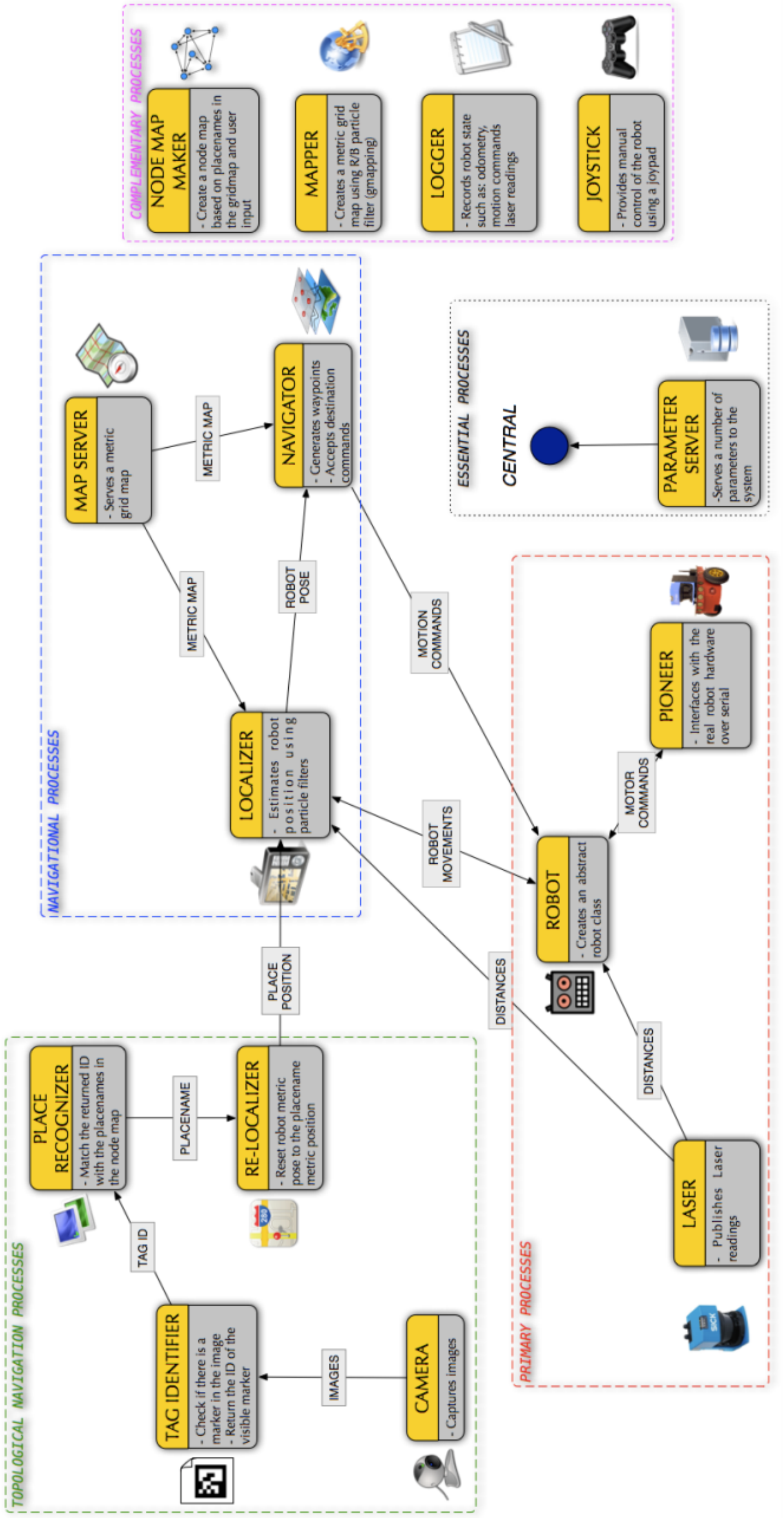

Figure 11, The developed approach

Chapter 4

Papers and Extended Work

The rest of this thesis has been organized as a collection of papers. These papers have been reformatted and form individual pieces of work at the end of the thesis.

This chapter aims to highlight the findings that are presented in these papers. Moreover, this chapter also presents the extensions of these works, which are not included in original papers. These extensions are either made after the publication/submission of papers or could not be included in the papers at the first place, due to space considerations.

This thesis is the result of an Industrial PhD project. Therefore, in addition to scientific papers, physical experiments and proof-of-the-concept demonstrations play an important role in dissemination of the knowledge. In line with this goal, a functional prototype is developed.

Following several experiments, the prototype was demonstrated at the Bispebjerg Hospital in several occasions; with the participation of several stakeholders from the hospital, DTU and Force Technology. In addition to being very useful to show many key concepts that were developed during the project, the demonstrations gave us the opportunity to grasp the reactions and expectations of prospective users. A short video, compiled from these demonstrations can be seen at http://bit.ly/hwo0Sx.

4.1 Service Robots for Hospitals: A Case Study of Transportation Tasks in a Hospital

The Industrial PhD project aims to outline the design of an automated logistics system for hospitals. Accordingly, one of the fundamental tasks before delving into the technical issues is to analyze the existing transportation system in a hospital.

This paper concisely deals with this task: Taking a snapshot of the transportation system at Bispebjerg hospital and outlining the design guidelines for robotic transportation.

Such an analysis is not only important for the specific task of developing transportation robots for Bispebjerg hospital; but it gives an insight to the hospital logistics, and provides a significant resource to designers, developers and researchers of service robots. Consequently, the main contribution of this paper is the analysis itself, and the design guidelines for implementing robotic transportation systems in hospitals.

The summary of the transportation tasks is given in Table 2. In other words, the table shows the vast amount of resources that are should be canalized to logistics; in order to keep up the healthcare service in a typical hospital.

Table 2, Summary of transportation tasks in Bispebjerg Hospital

Task	***Personnel***	***Hours per Week***
Medicine Transportation	Porters (2)	50.4
Mail Delivery	Hospital Employee (4)	107.1
Refills for Storages in Units	Porter (1)	27
Refills for Sterile Cabinets	Hospital Employee (2)	69
Meals and Food	Porters (5)	98
Transportation from Central Supply (non-medical)	Porters (2)	51,8
Clothes and Laundry	Porters (4)	130
Waste Collection	Porters (4)	195.84
Clean/Dirty Beds	Porters (3-4)	113.4
Total:	***Min. 27 Personnel***	***842.54 hours***

Automation of the logistics is a broad concept and service robots can be implemented in various levels. In accordance with the analysis, three scenarios are developed for possible implementation of robotic transportation:

1. Adaptation of service robots to the existing logistics system
2. Partial reconfiguration of the logistics system
3. Reconstruction of the logistics system

These scenarios are in the order of increasing complexity; while a total reconstruction of the system will result in the highest efficiency, it is also the most costly option. Similarly, replacement of human porters with service robots is the simplest, yet a sub-optimal solution. This paper therefore suggests a partial reconfiguration of the logistics system as the most optimal way to introduce service robots to the hospital. In this scenario, transportation tasks are prioritized and redefined for automation. Existing storage facilities are utilized, but transportation routines can be stretched to 24 hours.

In May 2010 (after the publication of this paper), Bispebjerg Hospital is granted funding for renovations under the regional healthcare reform. The project will be finalized in 2020, with an addition of 127,100 square meters to the existing hospital.

Regarding the renovation project, a similar analysis is conducted by Bispebjerg Hospital, with the aim of analyzing the usage of the tunnel network (Figure 12) that connects the majority of the buildings in the hospital [111]. Over 14 days, the traffic at tunnels was recorded. Similar to the paper, the goal of this new analysis is to provide input to the renovation project.

Results again show the vastness of the traffic and material flow inside the hospital: Over 31,000 people were recorded, with an average of 2,000 users per day. Busiest days are Wednesday and Thursday, whereas the highest traffic occurs between 9 and 10 o'clock in the morning as seen in Figure 13.

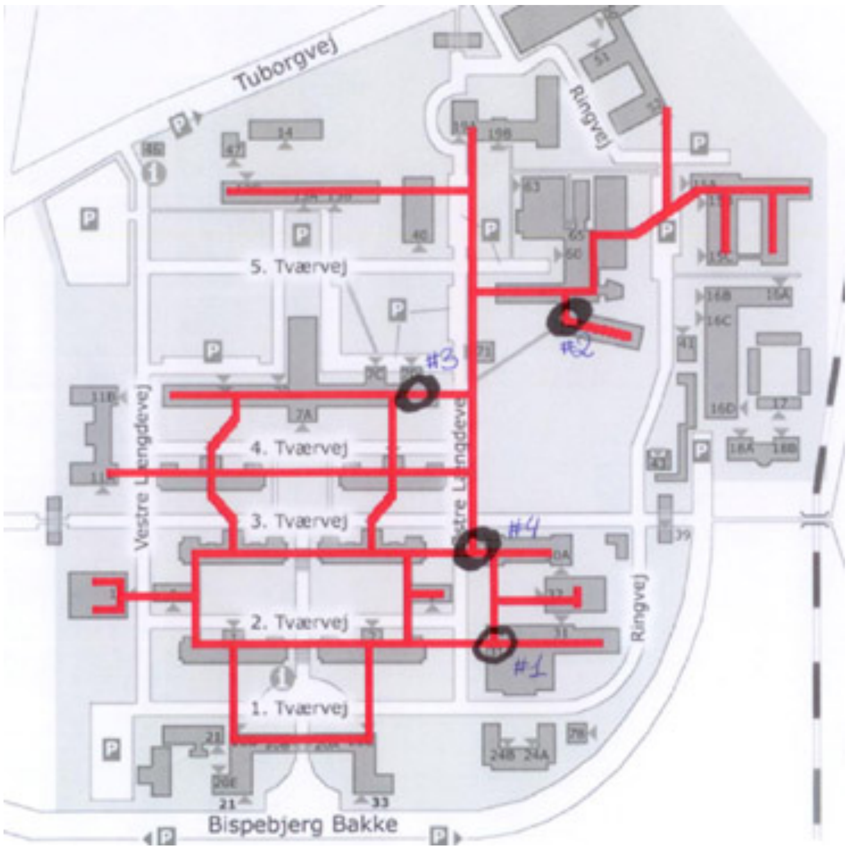

Figure 12, The tunnel network of Bispebjerg Hospital. Marked locations were used for sampling the traffic.

The analysis also depicts the profile of the users. As expected, porters are the ones that use the tunnels most frequently and they are followed by nurses, and postal workers.

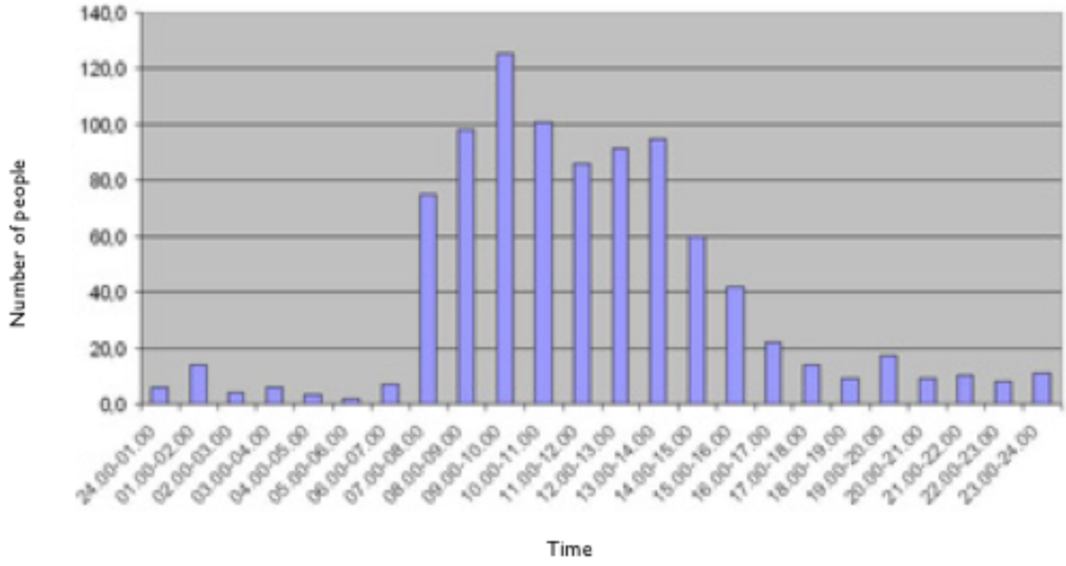

Figure 13, Daily usage of the tunnel network in Bispebjerg Hospital [111]

4.2 Mobile Robot Navigation Using Visual Tags: A Review

Reliability, scalability and adaptability are three key concepts in this PhD project. Several design considerations were made to achieve a robust system, including the use of hybrid maps; which is in turn related to the concept of 'nodes' and 'topology'.

To achieve highly reliable and multi-functional topological maps, it is necessary to mark certain areas of the environments. In this paper, a review of the visual tags is given, with the intention of finding the most suitable types of visual tags for service robot applications. Visual tags are chosen over other types of markers such as RFID tags or infrared beacons, due to their ease of installation, very low cost and simplicity (they are in essence, regular print-outs).

Due to their long range of use, a number of visual tags have been developed. In this paper, they are categorized as 1-D/2-D Barcodes and template based / id based / topology based augmented reality markers. While barcodes mainly focus on data storage, augmented reality markers aim easy detectability. In short, augmented reality markers are designed for estimating the relative pose of the camera, and augmenting computer generated 3D objects into the live video stream.

Looking at the usage of visual tags in mobile robot navigation; a long list of systems and applications can be observed. Many of these systems are proprietary and solely designed for the specific task. These tasks include visual servoing, identification and pose estimation.

Reviewing the literature in robotics reveals the common problem of '*reinventing the wheel*' and it also underlines the purpose of this paper. Instead of designing a new type of visual tag that can have only a limited use in the context of the service robot application, it is aimed to identify the most suitable visual tagging systems that have wider usage and established methods for encoding and decoding.

As a result of the review two types are identified as the most suitable types of visual tags for mobile robot navigation. Id-based augmented reality markers are suggested, due to the fact that they are already designed for real-time detection with depth estimation capabilities. This makes them suitable for robotic systems that typically have limited computational resources. In essence, they are called Id based because each marker points to a number between 0 and 4095.

2D barcodes (particularly QR Codes and Datamatrix) are also found to be potentially useful for robotic applications. They are already being used in pervasive computing applications, and their main advantage over augmented reality barcodes is the possibility of encoding any type of information directly on the barcode. This opens the door to building semantic topologies, where relevant information about each node is directly stored in the visual tag that is attached at it's physical location. On the other hand, detection of 2D barcodes is typically slower, and they are not originally designed for estimating depth.

4.2.1 Simulations

Based on the conclusions of the paper, two simulation scenarios are developed to test the detection and pose estimation characteristics of augmented reality markers. Using PovRay [112], a programmatic 3D tool, translation and rotation simulation data are generated.

In the translation test (Figure 14), camera simply moves above the marker in horizontal direction. As the marker becomes fully visible (around frame no.20), pose estimation becomes possible. Distances (x,y,z) are estimated quite accurately; errors are bound within 0.2 cm. Roll angle is estimated as 180 degrees, and Yaw angle is found to be 0. Most fluctuations are observed in Pitch angle (max. 8 degrees), whereas the actual value is 0 degrees.

Similarly, the camera rotates around the marker with a fixed radius of 40 cm (Figure 15). The results show that depth estimation is orientation dependent: As the marker becomes perpendicular to the camera, estimations become more accurate. Angles also suffer from orientation, the detection becomes possible between 25-165 degrees.

In general, estimation errors are found to be non-critical (small compared to mapping/localization errors of a robot), and can be further eliminated by data smoothing.

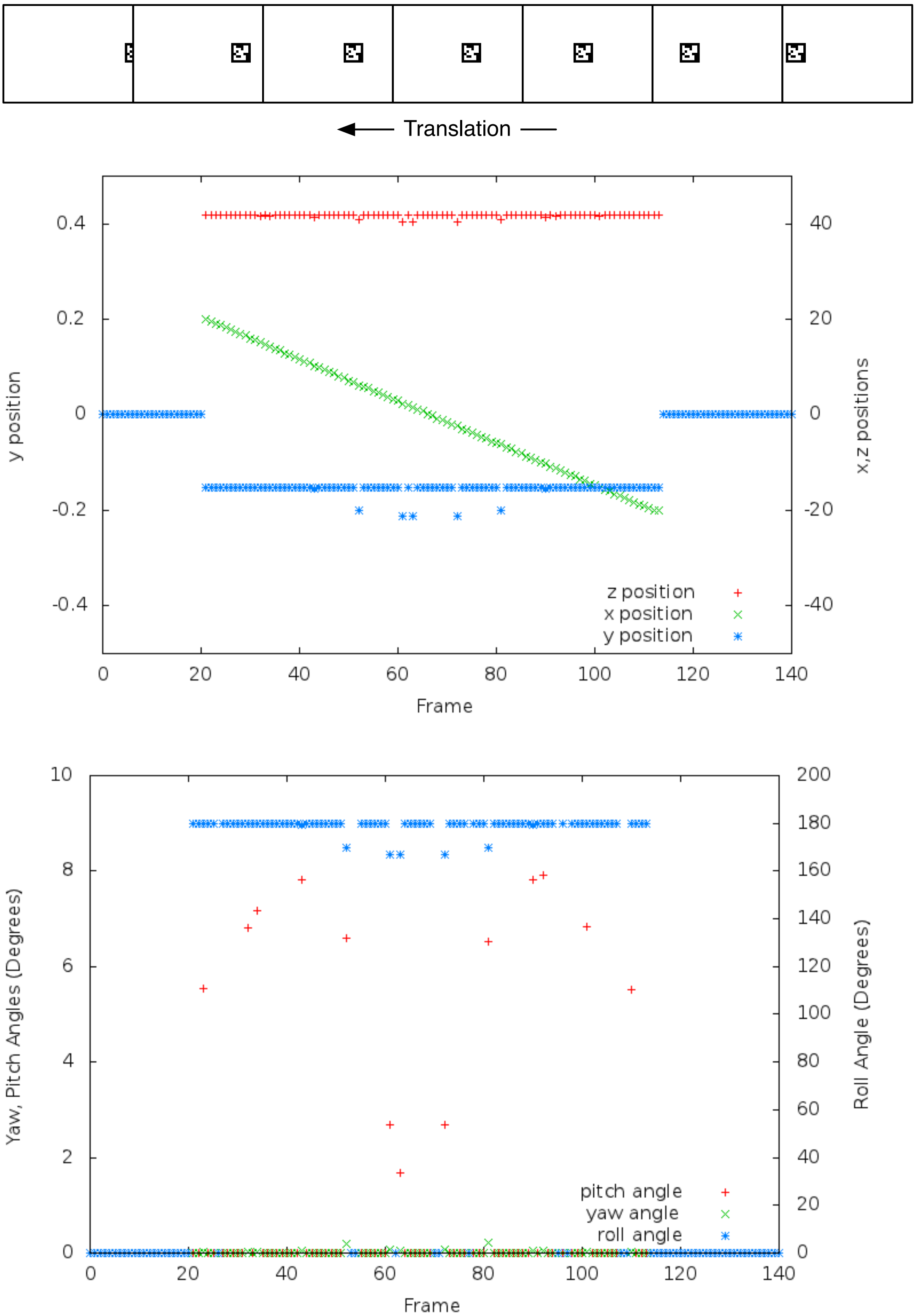

Figure 14, Translation test. Camera moves over the marker in x direction (top). The elevation of the camera is estimated as 40 cm, whereas the x position of the camera varies between 20/-20 cm. (middle). Roll angle is estimated as 180 (bottom).

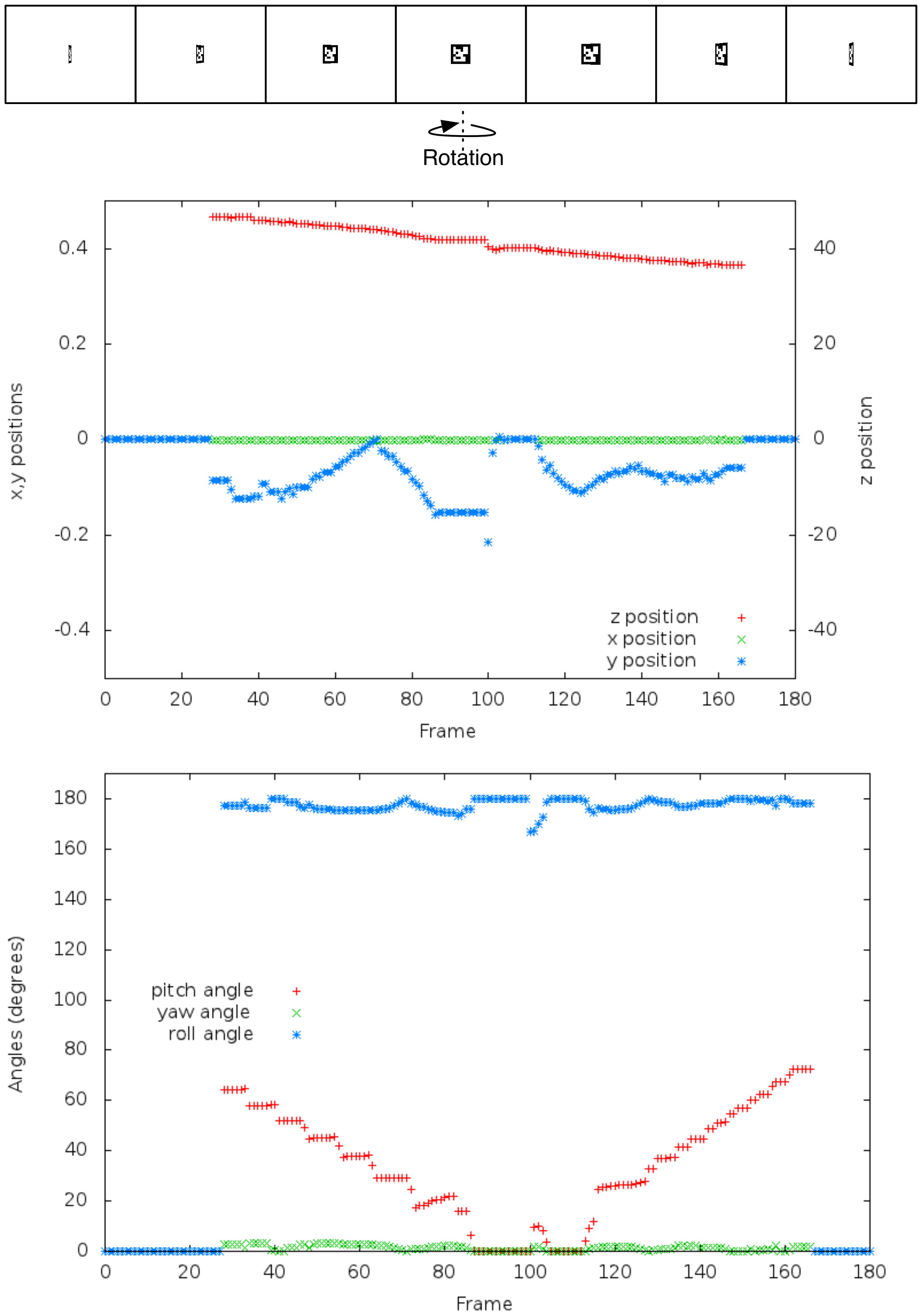

Figure 15, Rotation test. Camera moves around the center axis of the marker (top). The elevation of the camera (radius of rotation) is estimated around 40 cm (middle). Marker detection starts at an angle of approx. 70 degrees. (bottom)

4.3 Practical Indoor Mobile Robot Navigation Using Hybrid Maps

This paper aims to outline a framework for the design of service robots in hospitals, and it proposes a generic method for navigating in typical indoor environments. It approaches the problem from system design perspective, and provides a reliable, scalable and adaptable solution.

Despite most of the work that focus on mapping, localization and path-planning problems separately, this method provides an integrated solution that affects how the environment is represented and how the solution can be implemented in real life.

The key concept in this work is hybrid map representation; that allows managing complexity. Instead of mapping the whole environment in a single, globally consistent map (as in Figure 19); our method only deals with small local maps, which are easy to obtain (Figure 20). These slightly overlapping local maps are manually annotated – an inevitably manual process to augment semantic information, and based on the correlations of the nodes (annotations); a global topology of the environment is achieved (Figure 16).

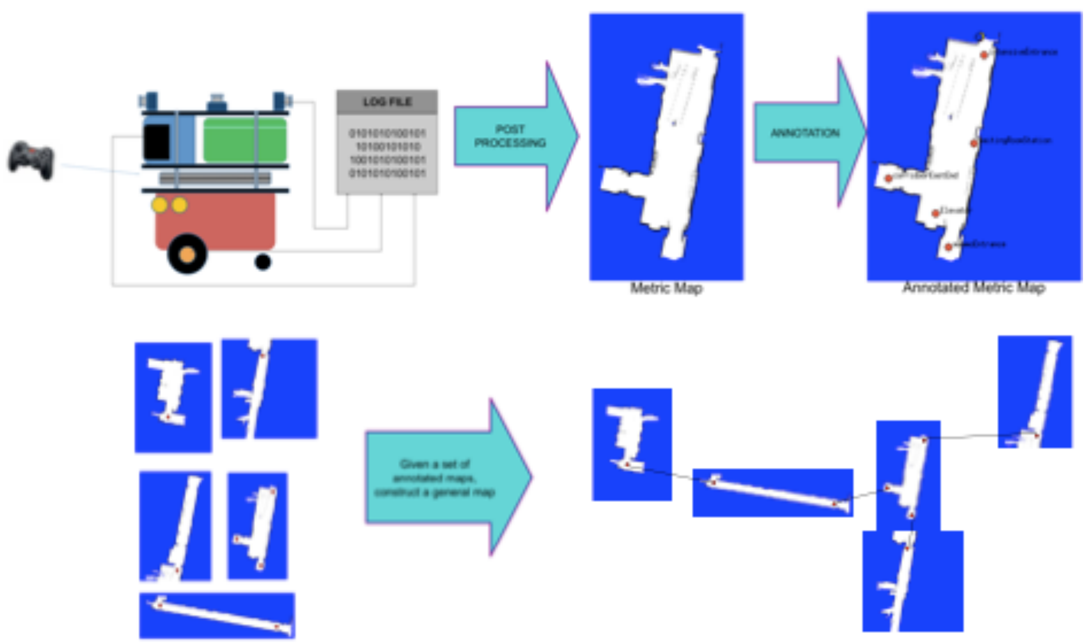

Figure 16, Summary of hybrid mapping. The robot is first manually driven in the local environment to record a dataset. Then the dataset is processed to obtain a metric map; which is in turn annotated. Given a set of annotated local maps, the method finds the commonly occurring nodes and creates symbolic links between local maps to obtain global topology.

Having this representation, it is easily possible to plan a topological path from robot's current destination to a point-of-interest. During normal use of the robot, users are only prompted to command the robot to a destination. The available destinations are automatically extracted from the global topology map, and they abstract numeric pose information of the points-of-interest in a given local map (e.g. "Steen's Office" correspond to x=2.453m, y=3.425m, θ=0.297rad).

Places-of-interest are marked with visual tags (id-based augmented reality markers), in order to create a redundant layer for topological localization and mapping. Using the markers, it is possible to identify switching nodes (where local maps overlap).

The overall advantages of the method can be summarized as follows:

1. Manageable mapping: Obtaining metric maps of large environments is rather difficult, if not time consuming. Mapping depends on simultaneous localization and mapping; where the path of the robot is estimated using odometry and corrected using range sensors. As the environment gets bigger and the robot needs to traverse large distances, errors in estimation and correction eventually grow and result in inconsistencies (Figure 17 illustrates this phenomenon). Partitioning the environment into chunks makes it possible to generate small, local maps, since global consistency is not necessary.

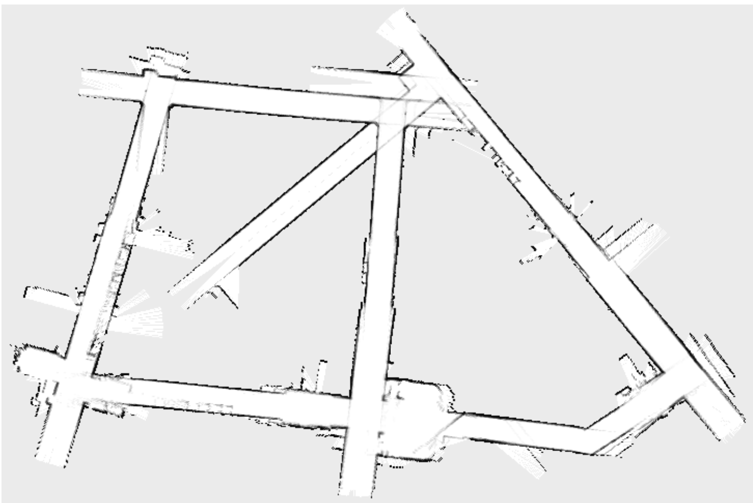

Figure 17, An inconsistent map of a relatively large are from Bispebjerg Tunnel Network. In addition to the size of the environment, several factors, such as un-even floor and slope differences result in an unusable map.

2. Adding maps over time: Mapping a large environment in one shot is often not possible. In addition to the reasons related to post-processing that are stated above, it is not practical or feasible to map such an environment due to the time it will take to traverse the environment. Again, from system design point; it is much more desirable to do it in small batches over time, or by using multiple robots. As

the method only deals with the relative topology of local maps; it is possible to systematically build the map of the whole environment step by step, as the new maps become necessary. Moreover, the existing maps can be altered in the same way, if a part of the environment changes due to e.g. construction. Similarly, new points-of-interest can be added modified or removed if necessary. These changes will be handled automatically, hiding the complexity from the user.

3. Vertical topology: Majority of the hospital buildings has multiple floors – an issue that is hardly touched in robotics research. Since this method only deals with global topology, it is also capable of representing multi-floor buildings (as illustrated in Figure 18). The way local maps are handled is purely abstract and only denoting the connectivity of places. As a result, the method does not differentiate whether a local map is connected to another local map in the same floor, or in a different floor.

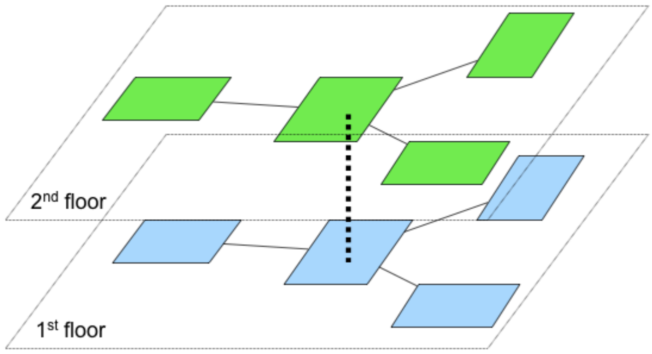

Figure 18, The method can transparently handle multi-floor buildings.

4. Simple path planning: Users of a service robot in a hospital does not necessarily concern with sending the robot to any given point in the environment. Instead, only a number of possible destinations are important. In this method, these places-of-interest, together with a number of 'switching nodes' are represented through a global topology map. Consequently, a global and abstract plan can be generated using this topology. This is also similar to how humans navigate; using landmarks as waypoints until the destination is reached.
Moreover, the environment representation used in this method makes it possible to create global topologies where nodes are connected with weighed links. Consider the city traffic, where certain paths are not optimal, even though they are shorter. Similarly, certain corridors in a hospital can be e.g. too crowded and should not be crossed at certain times of the day.

5. Simple localization: Due to the redundant representation of the environment, localization becomes more robust and reliable:

a. Any given time during navigation, only the map of a small part of the environment is active. Therefore, metric localization is only confined to a small local map.
b. Nodes of the topology map are marked with visual tags; which create a redundant layer for localization. Even though the metric localization of the robot might be off (it might loose it's accuracy for several reasons); observing a node (which is anchored on a local map) will re-calibrate the metric localization.

6. Semantic human interaction: Mapping, localization, and environment representation are inevitably robot-centric tasks. A robot needs to know positions and orientations of static objects (e.g. metric map) of the environment in order to plan its motions. On the other hand, humans require semantic information while using a service robot. Through hybrid maps, this method facilitates a common representation that can be used by both robots and humans. High-level and abstract global topology map is only available to humans as a simple list; and robots only deal with low-level, numeric local metric maps. Nodes act as glue between low-level and high-level maps, enable hybridization, and make the overall map useful to both humans and robots.

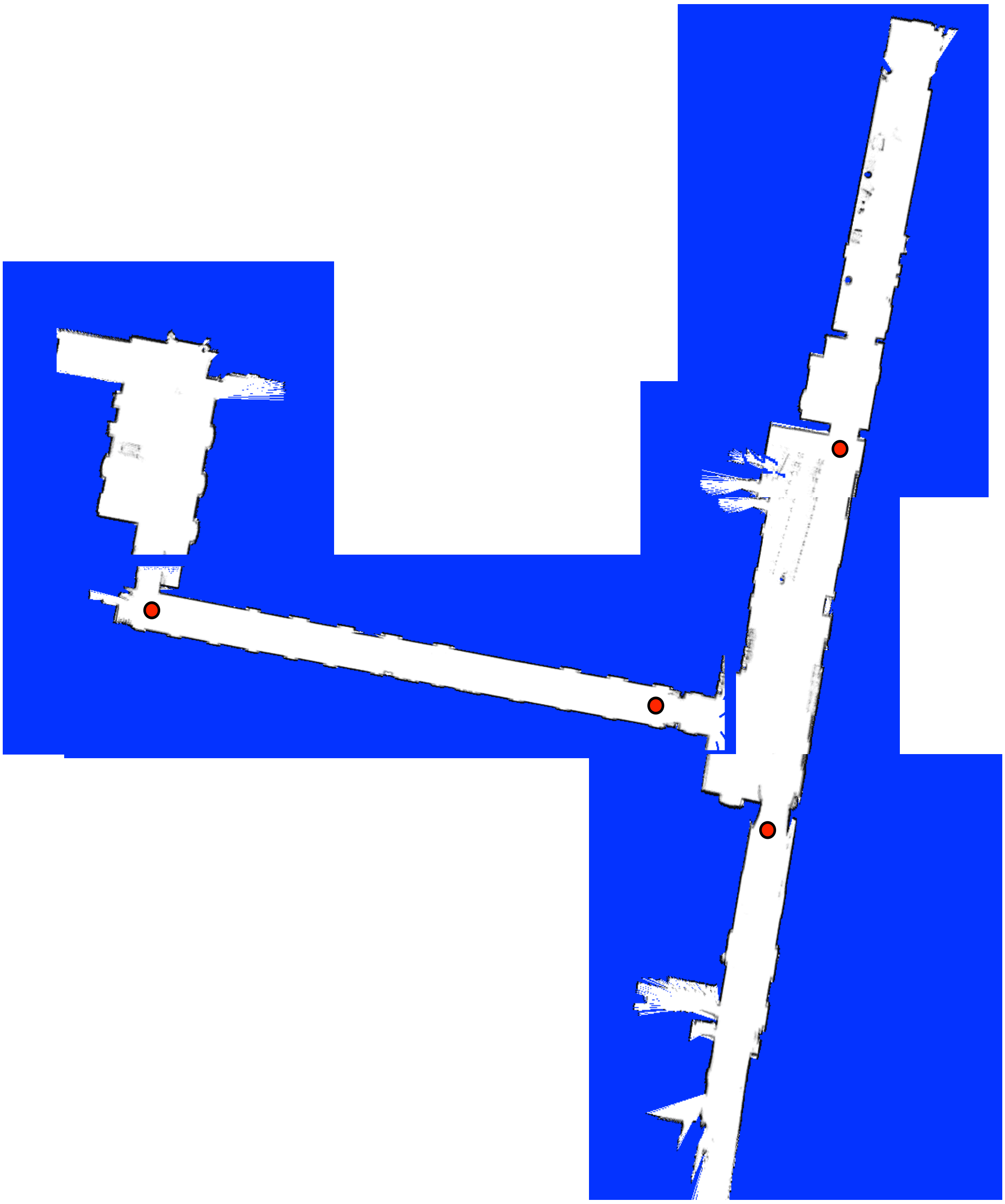

Figure 19, Actual physical layout of the environment. The maps are manually aligned, and the red dots represent "switching nodes". Obtaining such a layout of large environments or multi-floor buildings is not always possible or feasible.

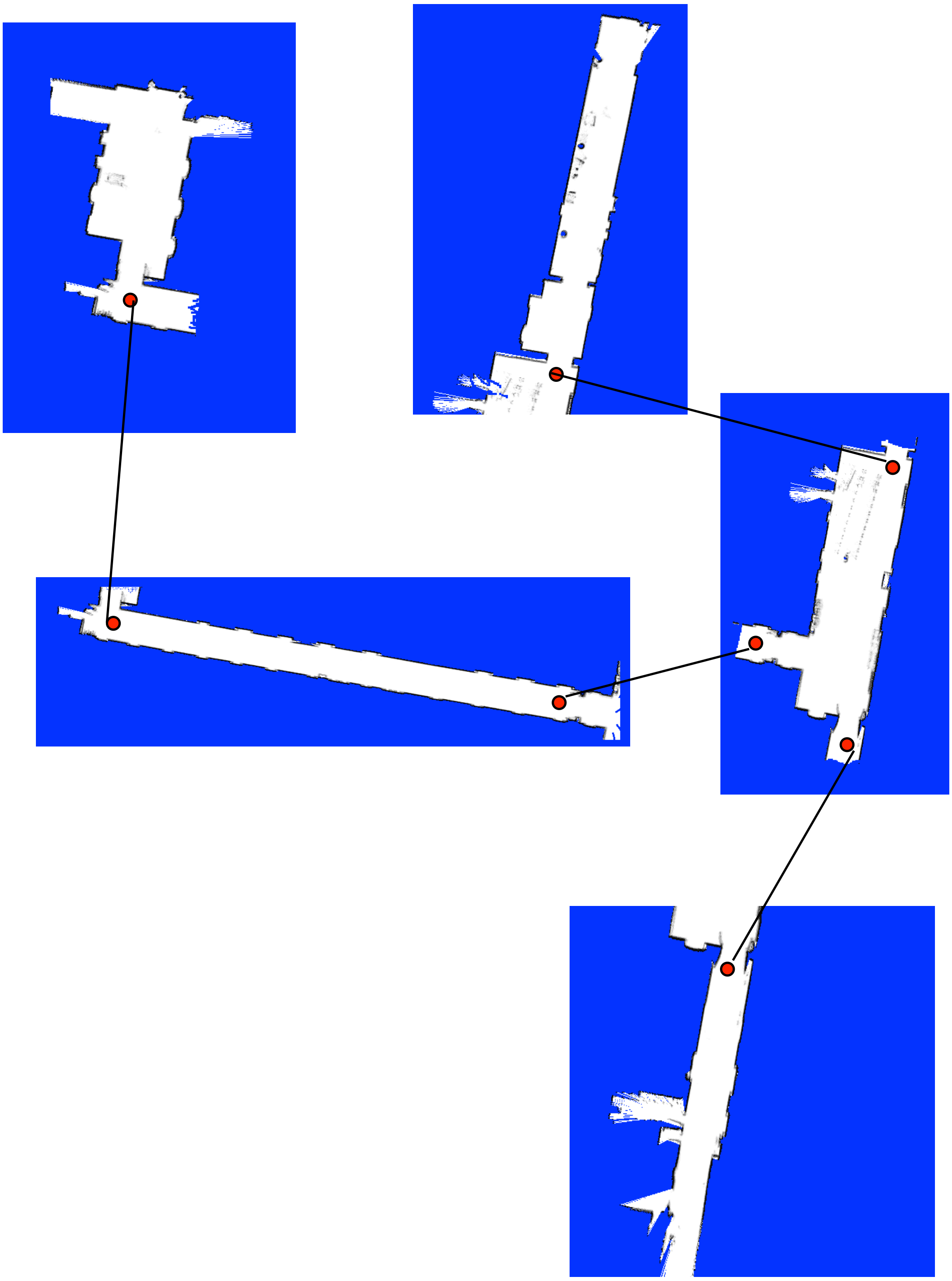

Figure 20, The same environment, represented with our method. The local maps do not need to be aligned, and therefore a globally consistent metric map. The local maps does not even need to be in the same resolution or orientation; as the global path only deals with the topology and the local paths are only planned within a single local map.

4.4 Automatically Annotated Mapping of Indoor Environments

This paper proposes a novel method to semantically annotate maps using 2D-barcodes. In relation to the previous paper, the aim is to create semantic maps for service robot applications with minimal human intervention.

2D-barcodes are already being used in numerous ubiquitous computing applications. They can visually encode information, and this information can be decoded by very simple hardware (e.g. camera-phones). Therefore, they can physically embed the semantic information to the locations they are attached.

In this work, we go one step further and exploit the physical characteristics of these barcodes to estimate their global positions in the environment. Since they can carry any data; along with the semantic information, we embed the physical characteristics of the barcode (i.e. only the width of the barcode, due to their square shape). Using triangulation, and four physically known points on a plane (Corners of the barcode, where the center of the barcode is the origin), it is possible to estimate the pose of the camera (therefore the robot), with respect to the 2D-barcode (Figure 21). Obtaining a number of observations (Figure 22) and a metric map with the help of range sensors, an automatically annotated, semantic map of the environment can be generated.

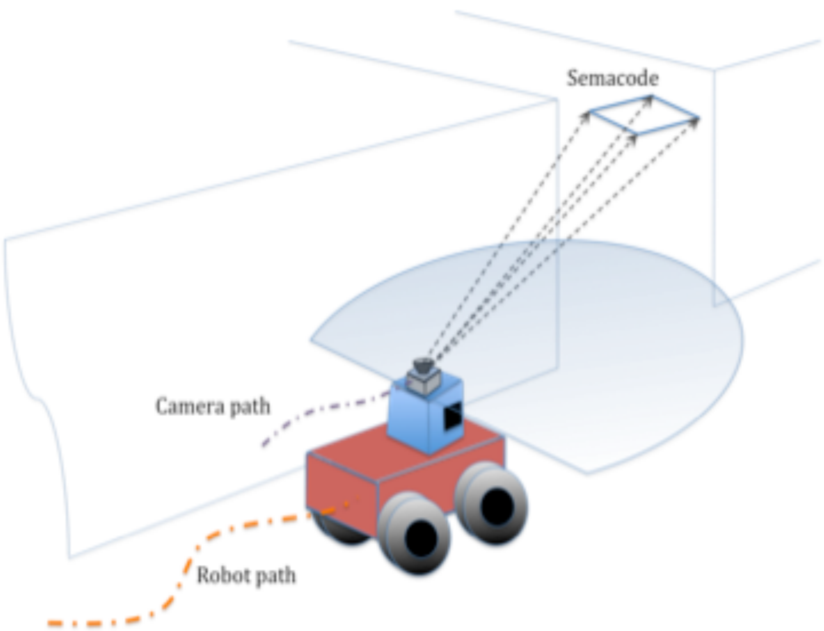

Figure 21, Fusion of two pose estimates: Range sensor is used to build a metric map, whereas cameras are used to extract topology and semantic information.

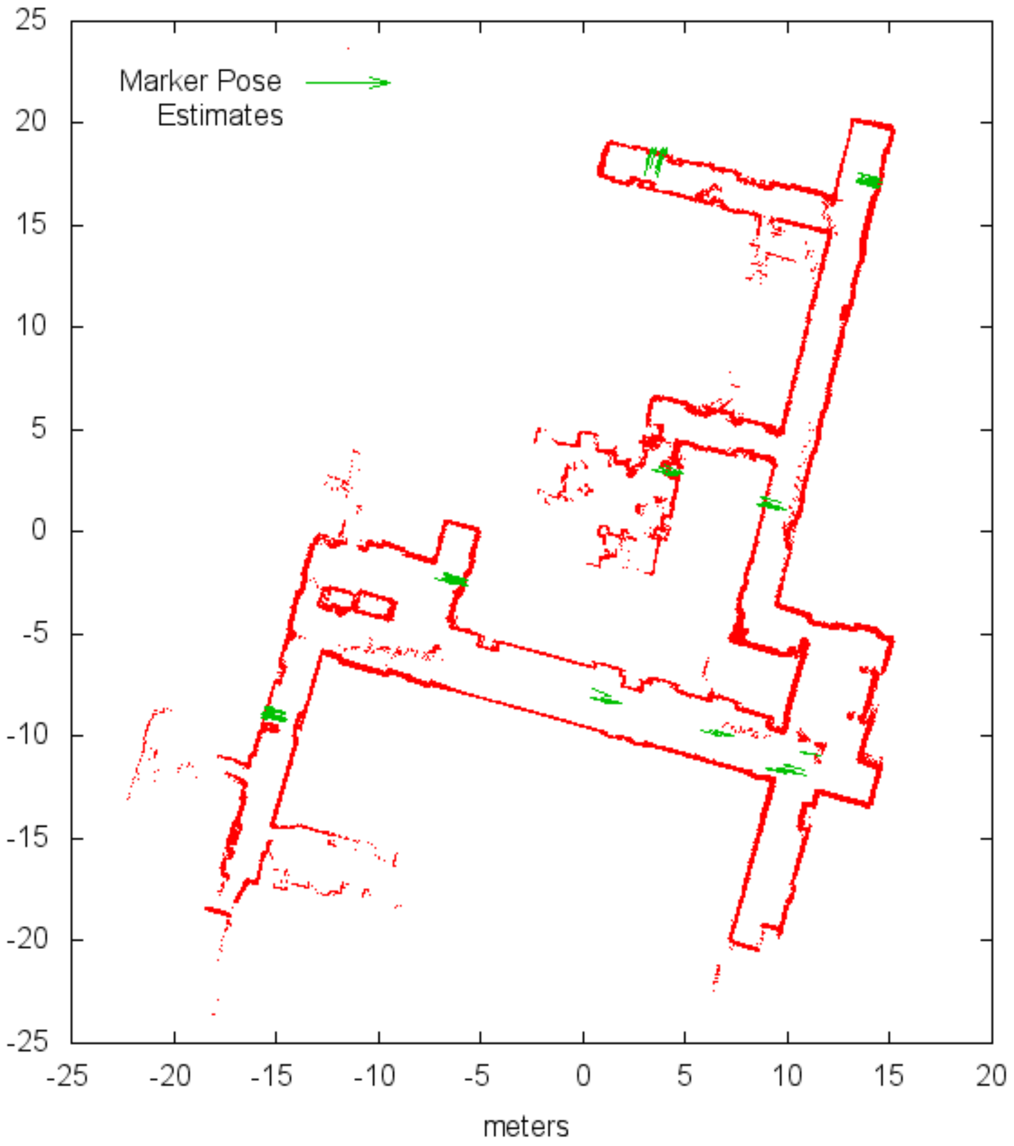

Figure 22, Raw estimates for marker poses.

In addition to the results presented here and on the paper; an interactive map can viewed at http://bit.ly/semacodedMapping.

The method presented in this paper is a novel way of annotating environments and creating hybrid maps. Places are physically linked, and marker poses are automatically estimated; making the method unique and very suitable for service robot applications.

2D barcodes used in this method are well-established international standards with a considerable user base. There already exists well-established and widely available methods for decoding 2D barcodes on any camera-phone; and these ubiquitous markers have already become a part of daily life in many parts of the world.

Therefore, the semantic representation presented in this work can be easily extended for multiple uses.

The video (which can be seen at http://bit.ly/whereAmI) shows a simple way of using the semantic map that is generated by the robot for human navigation. In this experiment, a camera-phone is used for localization and path planning inside a building. All of the computations are handled on the phone, making the solution stand-alone.

Figure 23, Doorplate design that incorporates Semantic Barcodes. These barcodes are so-called 'v-cards', that can carry a wide range of information, such as e-mail address, phone number, calendar information, website link.

These experiments are based on a scenario, where hospitals of the future incorporate pervasive computing technologies; having smart buildings with ambient intelligence. In accordance, a very simple modification on doorplates is proposed. As illustrated in Figure 23, a 2D barcode on the doorplate can provide vast amount of information to patients and hospital personnel. Hospitals are usually very large environments that impose difficulties to patients in terms of finding their ways inside the buildings. As illustrated in the video, our method can further utilize these doorplates for guiding patients.

4.5 Mapping of Multi-Floor Buildings: A Barometric Approach

This paper introduces an original method for mapping (and navigation) in multi-floor buildings. In terms of service robotics, vast majority of targeted environments are multi-floor buildings such as hospitals. Despite the crucial need, and despite the vast amount of research on mapping of single floors; the area of multi-floor buildings remains untouched.

The method presented in this paper exploits a simple phenomenon: Barometric pressure varies with the altitude. In simple terms, we do sensor fusion, by separating the range data from different floors, based on barometer readings from an inexpensive pressure sensor. As a result, we can automatically generate globally consistent 2.1 dimensional maps of multi-floor buildings. The maps are called 2.1D because each floor map is represented on a plane in 3D.

It is a known fact that the barometric pressure varies with atmospheric events, and therefore does not remain the same at the same floor in time. On the other hand, it is also a fact that the changes in barometric pressure over time occur relatively slowly, and without sharp fluctuations (Figure 24).

As explained in detail in the paper, this method is mainly concerns with the detection of floor transitions, where the robot travels from one floor to another. Considering the fact that moving from one floor to another using elevators takes at most a few minutes, the significant changes in pressure readings can be regarded as the change of the floor.

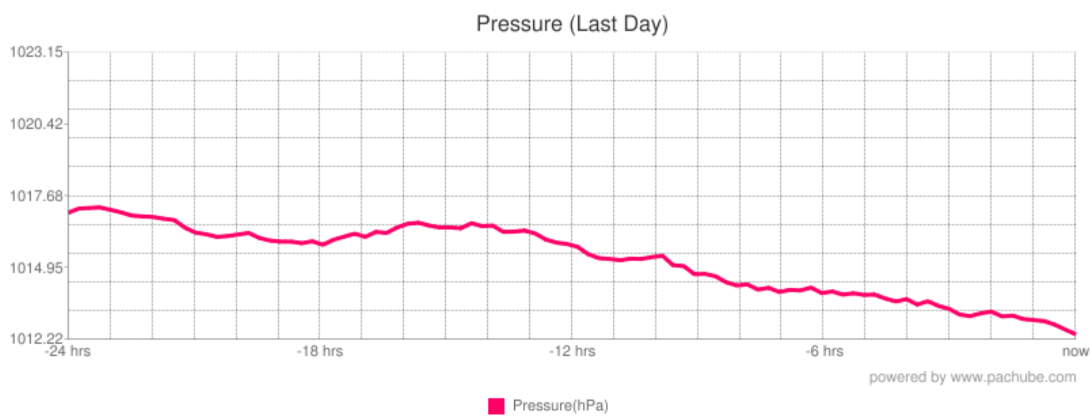

Figure 24, Pressure readings over 24 hours from a static location.

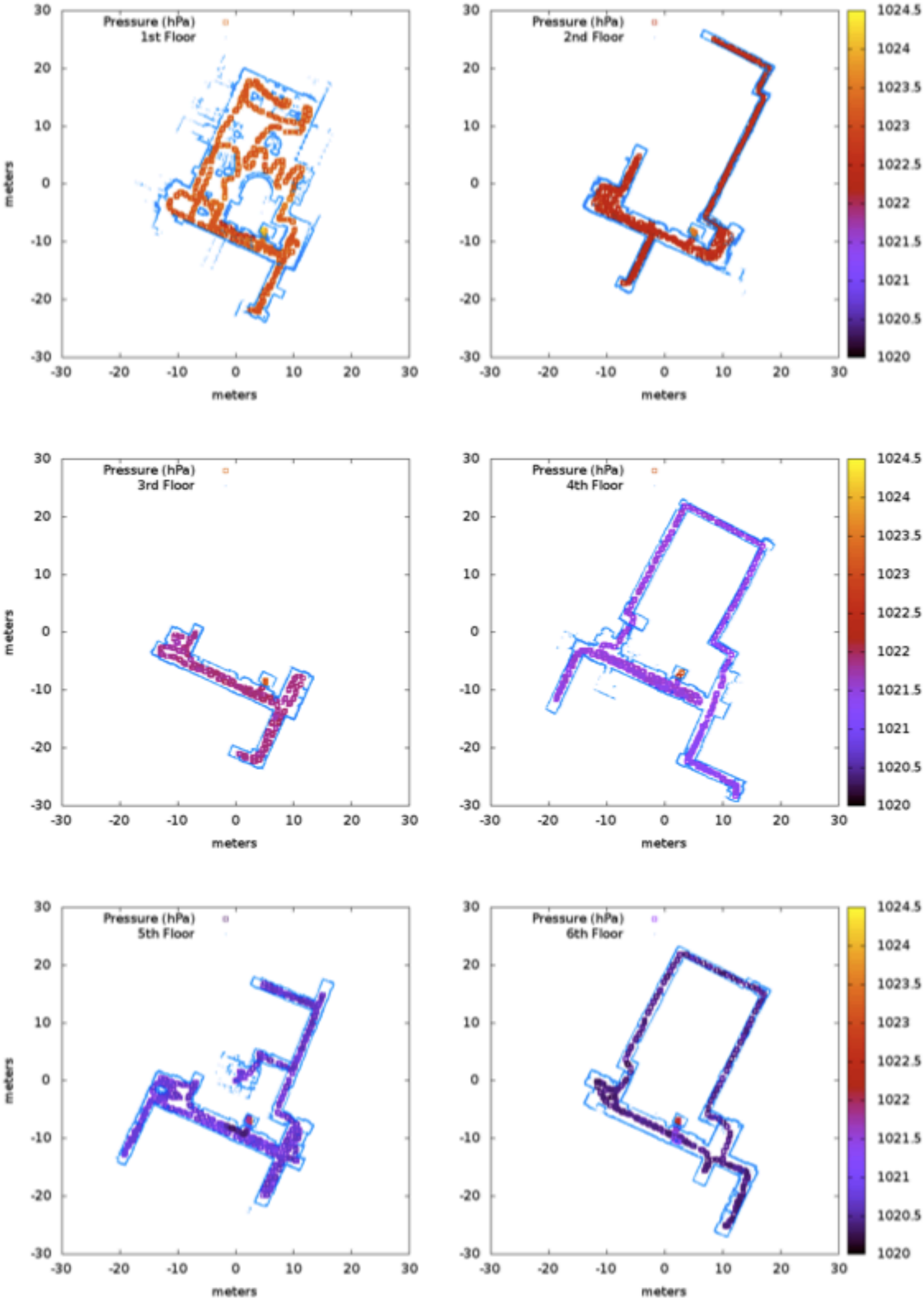

Figure 25, Pressure readings from 6 different floors of a building. It can be easily seen that the barometric pressure (represented in colors) is uniform and different than other floors. In the same way, it is also possible to spot the elevators where the pressure (color) is relatively different than the general pressure (color) of the floor.

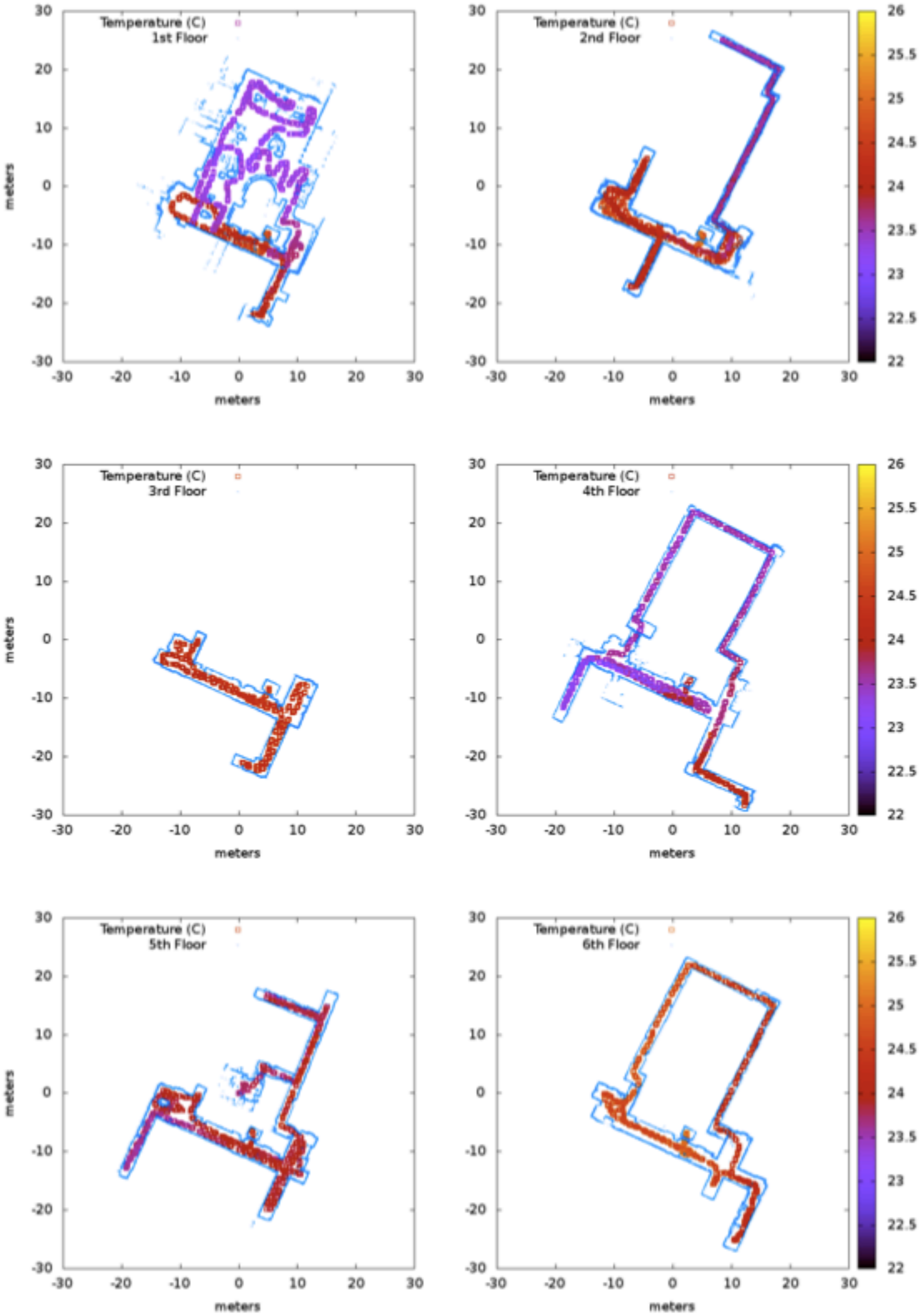

Figure 26, The temperature maps of the same building. It can be seen that despite the changes in temperature, pressure readings remain stable (previous figure). Even though it is not used for robotic purposes, these maps can provide very valuable information to building engineers; such as locating the heat losses in the building and assessing the efficiency of the heating, ventilating and air conditioning system

The principles of the method can be visualized as in Figure 25, where pressure readings from 6 floors of a multi-floor building are illustrated in varying colors. As shown in the paper (Figure 68), the pressure remains relatively constant in individual floors, during the course of mapping.

Inspecting the pressure (colors) in different floors also leads to a simple semantic knowledge about the environment: locations of elevators. The locations can be easily detected by observing the highest variation in color within the same floor (Figure 27).

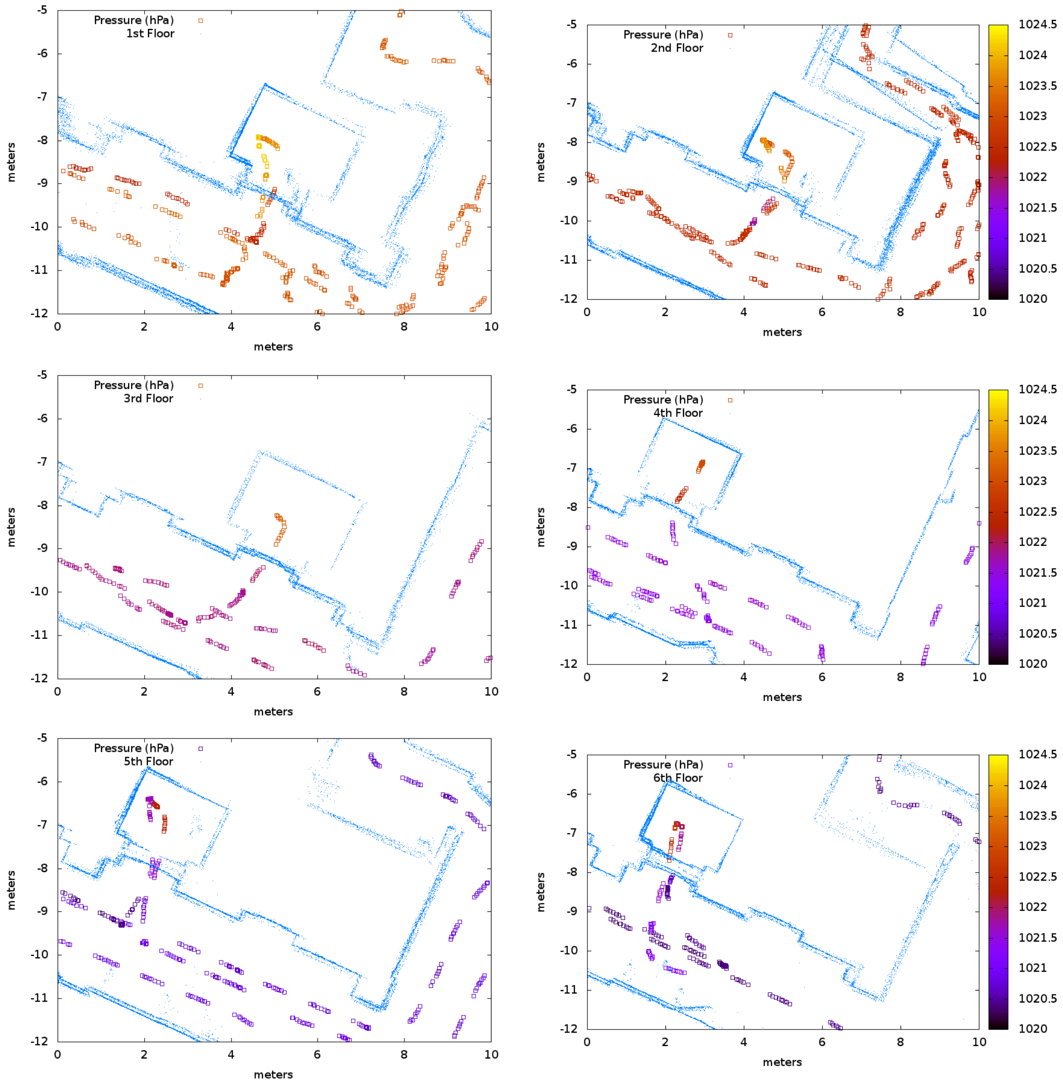

Figure 27, Elevators in detail. The pressure difference between the elevators and corridors are distinguishable. Moreover, it is also possible to see that two different elevators were used for transitions.

The method proposed in this paper addresses an important issue that is key to most service robot applications. It is developed around a scenario, where a single robot maps the environment in a single run; and the resulting map depicts the elevations of individual floors in terms of the barometric pressures at the time of mapping.

In addition to the method presented in the paper, we worked on a simple enhancement that extends the system: Having even a single reference reading from a know point will enable multi-robot mapping, vertical localization and globally consistent and time-invariant multi-floor maps. Since barometric sensors are inexpensive and they can be used beyond service robotic applications, it is even possible to install a sensor node in each floor.

This idea is further developed and a simple implementation is achieved. The sensor node we propose is evolved around a wireless access point. We interfaced a barometric pressure sensor to the internal board of the access point, and programmed the hardware to publish temperature and pressure data whenever requested (twitter feed for the sensor node can be seen at http://twiter.com/Reportweather).

4.6 Integration of Semantic Annotation and Multi-Floor Mapping Methods

Both automatic annotation and multi-floor mapping methods can be regarded as parts of a general framework. Therefore, it is obvious to integrate these two methods to obtain automatically annotated multi-floor maps. These maps offer a robust and unified representation to service robot applications.

In this context, an experiment is conducted at Bispebjerg Hospital. 24 places are designated as places-of-interest in two floors of the building 60, and they were tagged with 2D barcodes. The mobile robot is guided in the environment to collect images, range readings and pressure readings. The dataset were finally processed to generate the results that are presented in Figure 28 - Figure 33. Despite the lack of ground truth, it is observed that the generated map is very accurate, in terms of the annotations and their respective poses. Furthermore, maps are correctly split due to the barometric pressure difference between the floors (Figure 34).

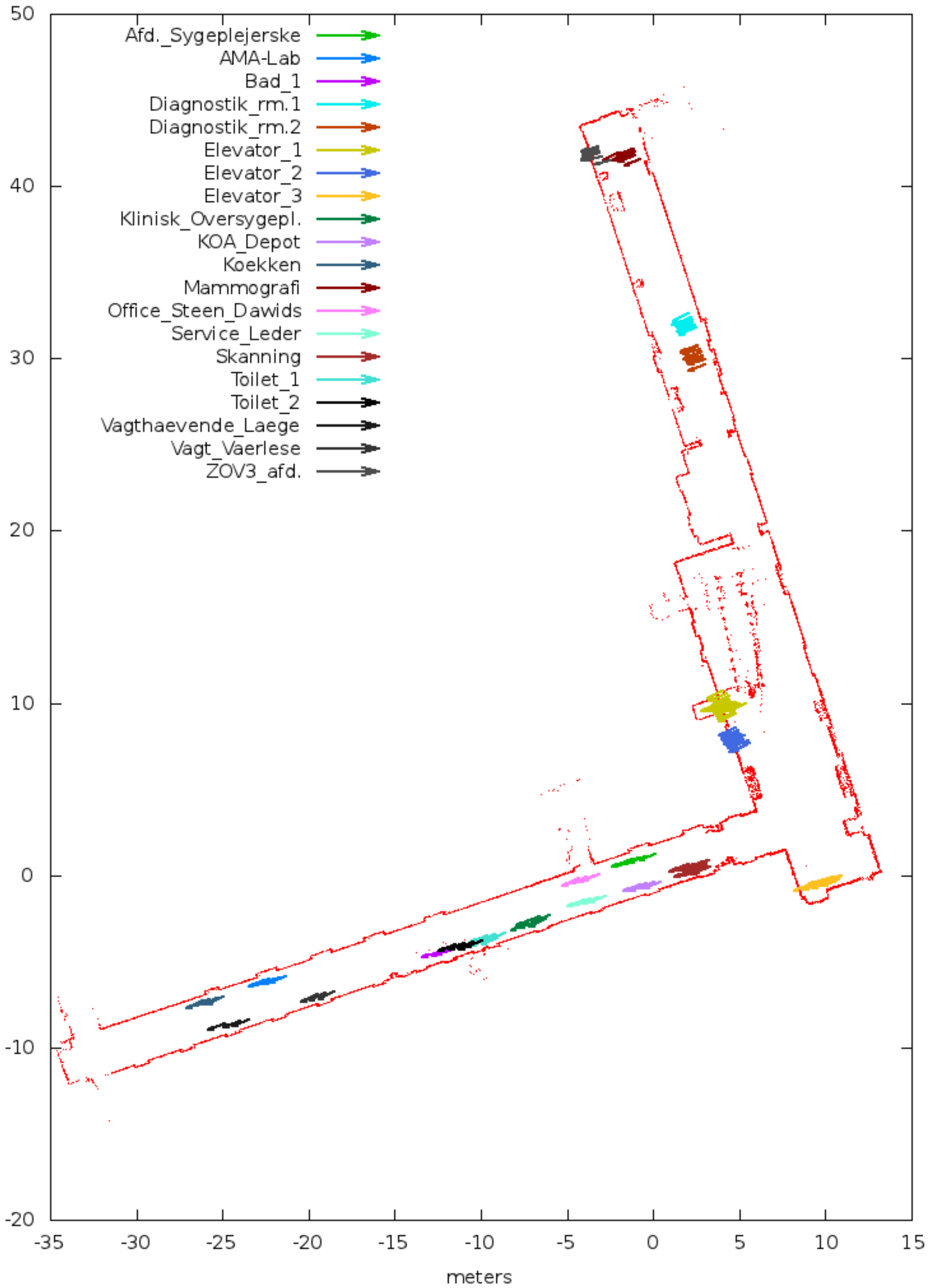

Figure 28, Marker pose estimates for 20 points-of-interest on the 1st floor of building 60 in Bispebjerg Hospital.

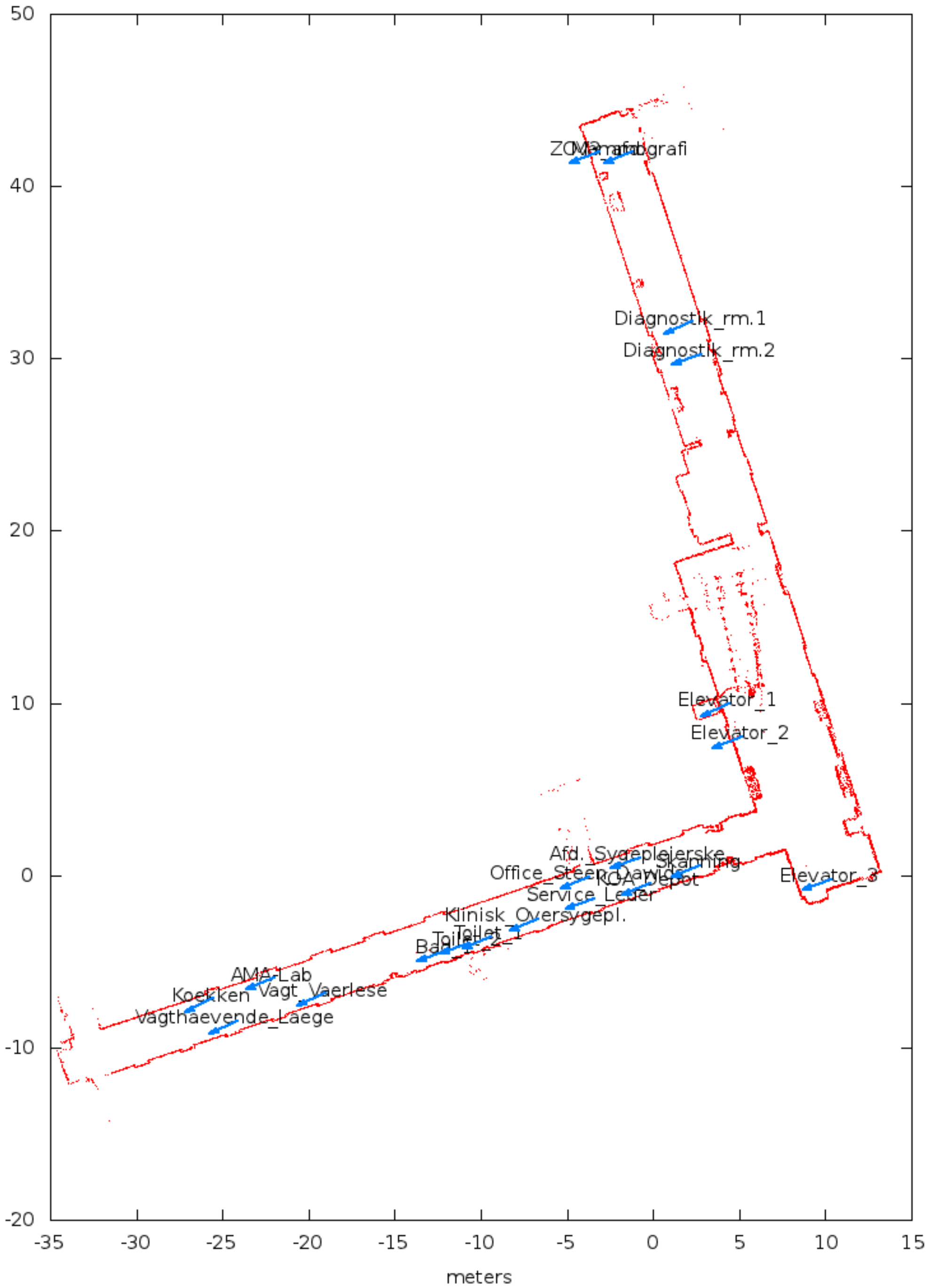

Figure 29, Annotated Map of the 1st floor

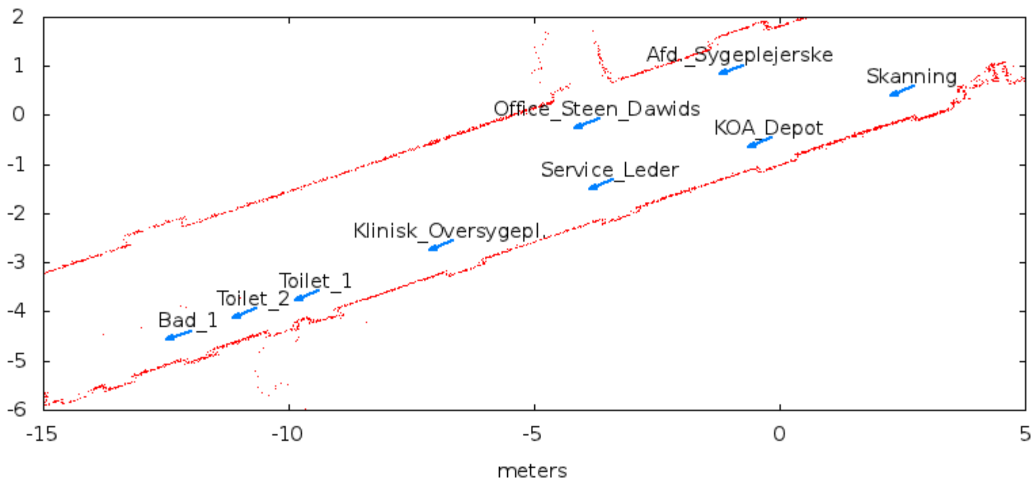

Figure 30, A close-up view from the 1st floor

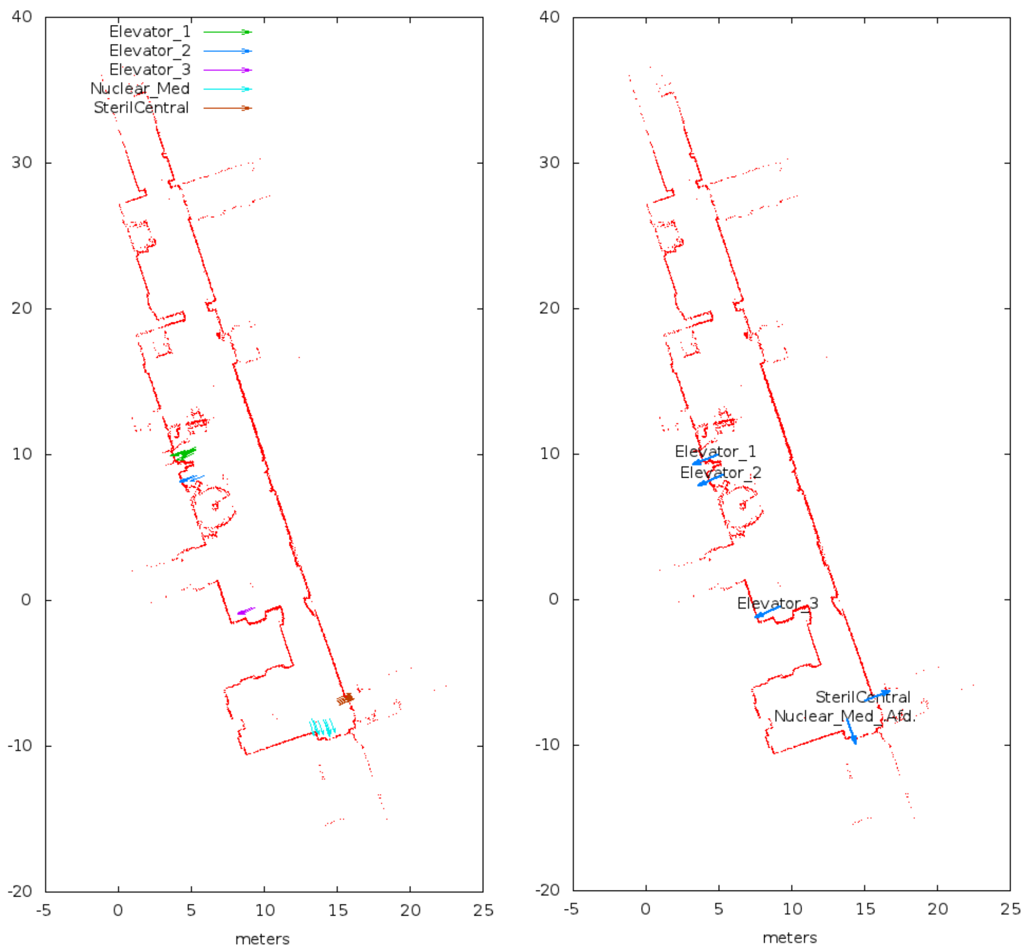

Figure 31, Marker pose estimates and Annotations from the ground floor

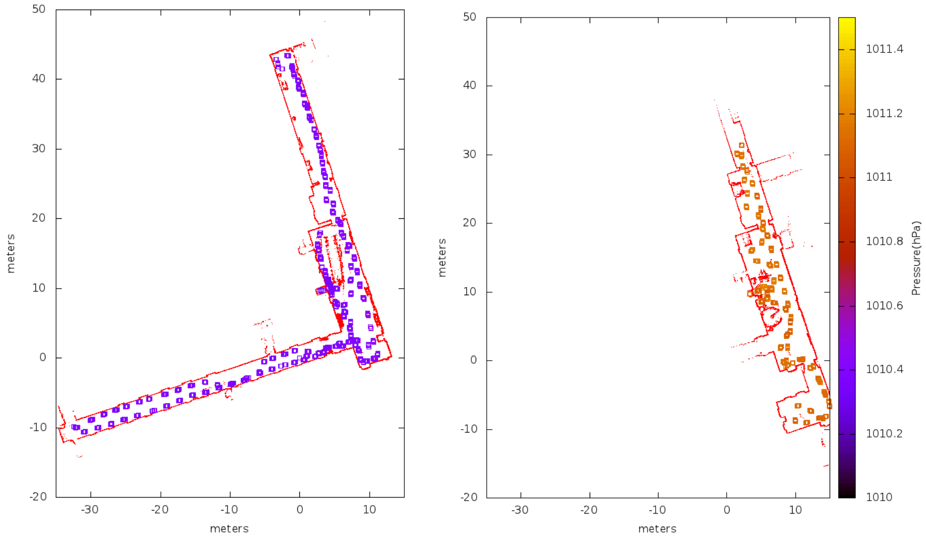

Figure 32, Pressure readings from 1st and ground floors

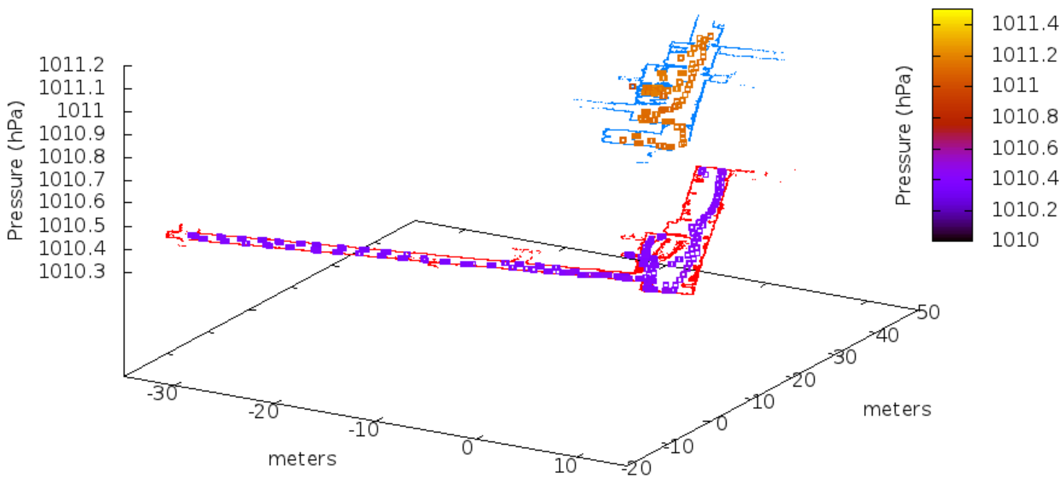

Figure 33, Correlated pressure maps from the ground floor and 1st floor.

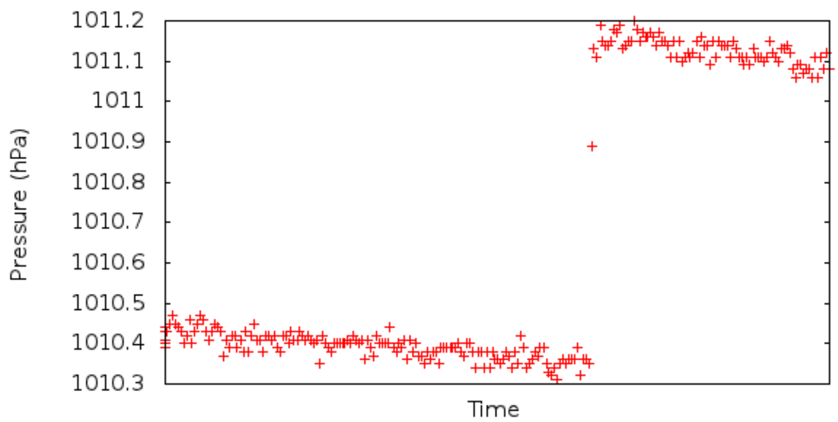

Figure 34, Pressure log during the experiment.

4.7 GPU-Accelerated Visual Localization

Localization is the process of estimating the pose of an *agent*, given a representation of the environment and sensor input. In this work, we extend the hybrid-mapping concept that is previously presented, by introducing *Topo-metric Appearance* Maps.

The main feature of *topo-metric appearance* maps is to represent the environment, based on appearances, without the need for visual tags. Such a representation also offers a new method for doing localization with a single camera.

With the introduction of Scale-Invariant Feature Transform (SIFT) algorithm [113], a number of vision-only localization methods have emerged. Commonly, these methods utilized local image detectors and descriptors (SIFT being the most popular) but utilized different strategies, such as building sparse-3D maps based on stereovision [114], Monte-Carlo localization [115], monocular simultaneous localization and mapping[116] and graph-based least square optimization [117]. Most of these methods, however, suffer from high computational costs of detecting, describing and matching features in images.

Utilization of graphical processing units (GPU) for general-purpose computation is a relatively new concept and highly active field of research. GPUs have massively parallel architectures that suit well for problems with simple repetitive computations. Consequently, several implementations of GPU based feature detection, tracking and matching are reported [118-121].

Based on the previous work that demonstrates the merits of hybrid representations, and the recently available GPU based image-processing techniques; we propose an appearance-based localization method that exploits *Topo-metric Appearance* Maps.

Environment representation is done by using a robot that is equipped with a laser range finder and an omnidirectional camera. The omnidirectional camera plays a key role in this method, as it offers a circular, 181-degree field-of-view *appearance* of the environment in a single frame. For localization, on the other hand, the only sensor that is necessary is the camera.

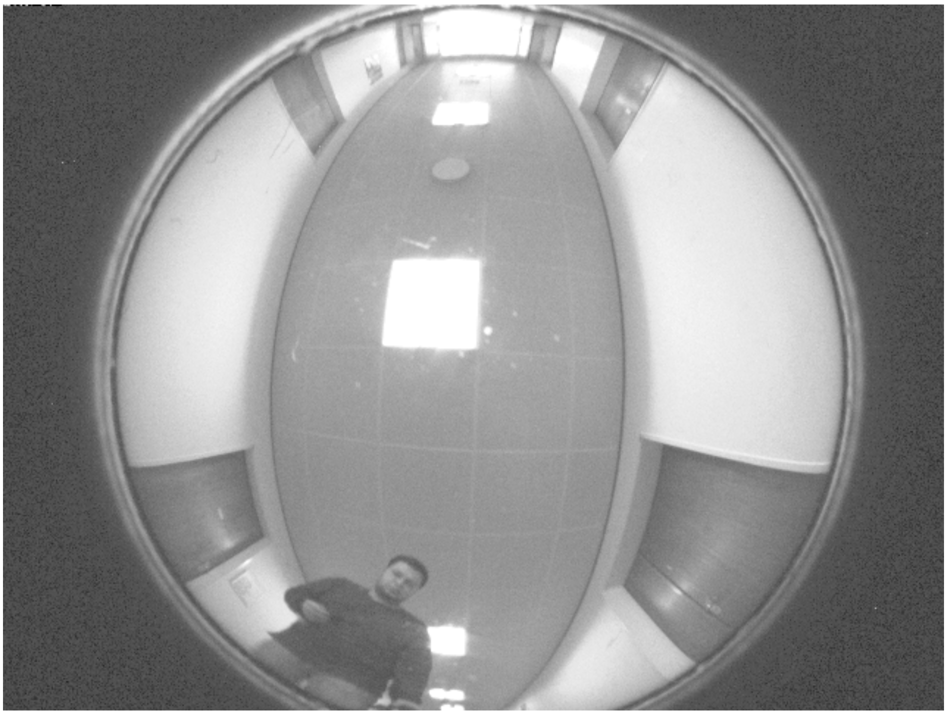

Figure 35, An omnidirectional image from one of the experiments in Bispebjerg Hospital. Using an inexpensive fisheye lens, it is possible to capture *appearances* of the environment using a single camera.

4.7.1 Method

4.7.1.1 Map Generation

Building a *Topo-metric Appearance* map is an offline process, and it requires a standard robot setup; which uses wheel encoders to estimate the path of the robot, a range sensor to detect objects in the environment and to correct the path estimation, and a camera to capture appearances (Figure 36). The path estimated by wheel encoders is prone to system noise and integration errors, and it needs to be corrected using a particle filter [122]. This process is usually referred as Simultaneous *Localization and Mapping* (SLAM), and it has two outputs: A metric grid-map and the corrected path of the robot during exploration.

Using the corrected path and the metric map of the robot, it is possible to select a sparse set of *appearances* from the dataset of collected images; and correlate them to metric poses based on their timestamps. The final result of this process is the *Topo-metric Appearance* Map*;* a sparse set of *appearances* with *known positions and orientations.* (An interactive topo-metric map can be seen at http://bit.ly/corridorView)

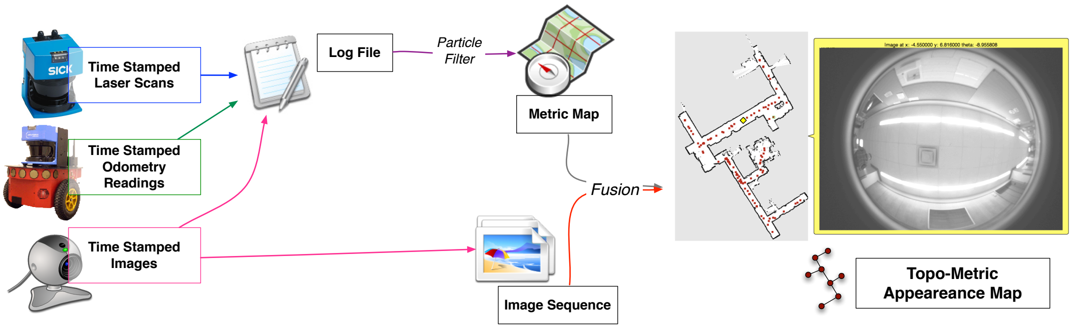

Figure 36, Making of a Topo-metric Appearance Map

4.7.1.2 Signature Generation

A *signature* is the quantitative characteristics of an *appearance*; whereas an *appearance* is a representative image of a scene in the environment. One way of creating a *signature* is to use local image features. In this work, we adopted *Scale Invariant Feature Transform (SIFT)* for detecting and describing features in *appearances.*

At this step, a vector of *signatures* is obtained, which correspond to the features of *all* images in the Appearance map.

4.7.1.3 Camera-only Localization

Having the *topo-metric appearance map* and the *signatures* vector; it is possible to do pose estimation, using a 'less capable' system; which might use a camera as the only sensor. The system simply compares captured frames during runtime to the images from the *appearance map*.

The overview of localization system is illustrated in Figure 37. This method relies on GPU acceleration [118]; live camera frame is transferred to GPU, its *signature* is computed, and matched against the *signatures vector* of *appearances.* Finally, the pointer to the best match is returned from the GPU, and the pose is estimated according to the *topo-metric appearance map.*

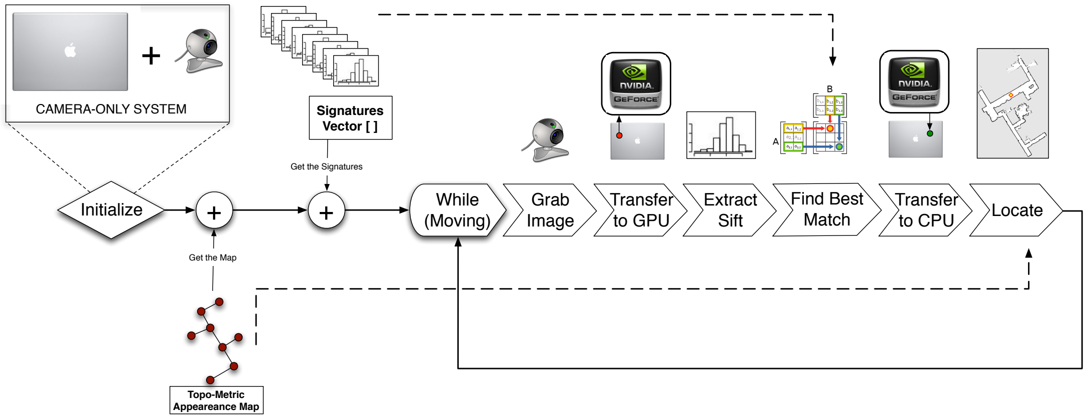

Figure 37, Camera-only localization using GPU

4.7.2 Empirical Validations

The method is first implemented on a mobile platform evaluated in a typical indoor environment. The 'capable' robot consists of a Pioneer 3-AT robot base, SICK-LMS200 laser range finder, a USB camera and a netbook computer (Figure 38, left). The robot is manually guided in an indoor environment (~30x40 meters), and the collected data is post-processed to obtain the *topo-metric appearance map,* which consists of 204 images (out of 915 collected) (Figure 39, right).

Localization performance is evaluated using a 'less capable' setup; which simply consisted of a USB camera connected to a laptop with discrete Nvidia graphics chipset (gt-9600). The setup is placed on a wheeled base (Figure 38, right), and manually moved inside the environment with fast walking speed (~1m/s). Best five matches from the *appearance* map was tracked, and it was observed that the system was able to consistently estimate its correct position in the map at a speed of ~1Hz, despite the changes in illumination, partial occlusions and elevation of the camera (Figure 39, left). More detailed results can be seen in the video that is accessible from http://bit.ly/cameraOnlyLocalization.

The presented method offers significant advantages that are not only limited to wheeled service robot systems. Since the only necessary sensor is a camera, the proposed method can be extended to a wide range of domains.

Unmanned aerial vehicles (UAV) are one of the ideal targets for extension, since the payloads of these flying robots are extremely critical. To investigate the usability of the topo-metric appearance maps for UAVs, we used the system shown in Figure 40. This helipad has four rotors, a micro-PC and a payload of approximately 500 grams. Using the wireless network, the helipad can relay the images to a stationary computer with a GPU.

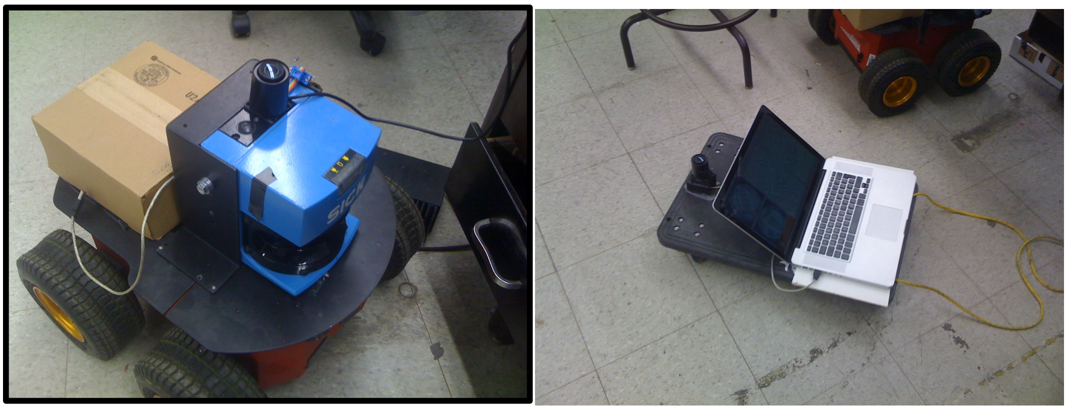

Figure 38, Platforms used for map generation (left) and for evaluating localization performance (right). The platform on the right is, in essence, a laptop and a camera on wheels.

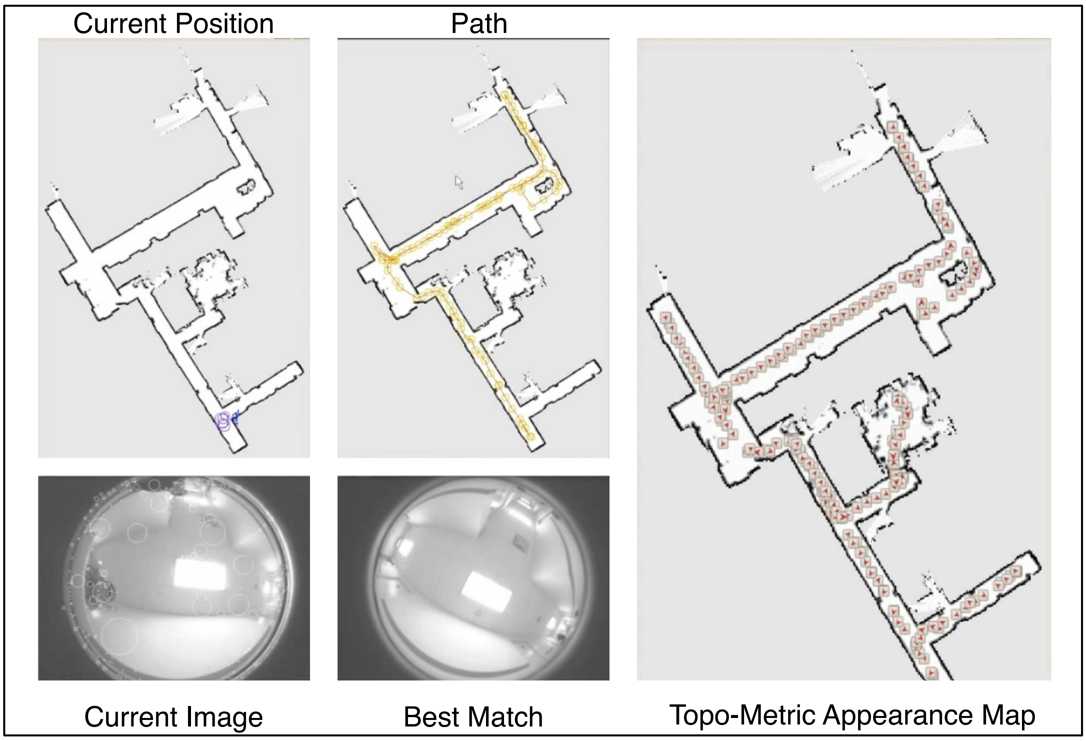

Figure 39, Localization using topo-metric appearance map.

During the experiment, the helipad is manually flied in the environment (for safety reasons). Images that are captured are transferred to the stationary computer, and signatures of *appearances* are calculated. Finally, using the same map that was generated by the robot in the previous experiment, localization is achieved.

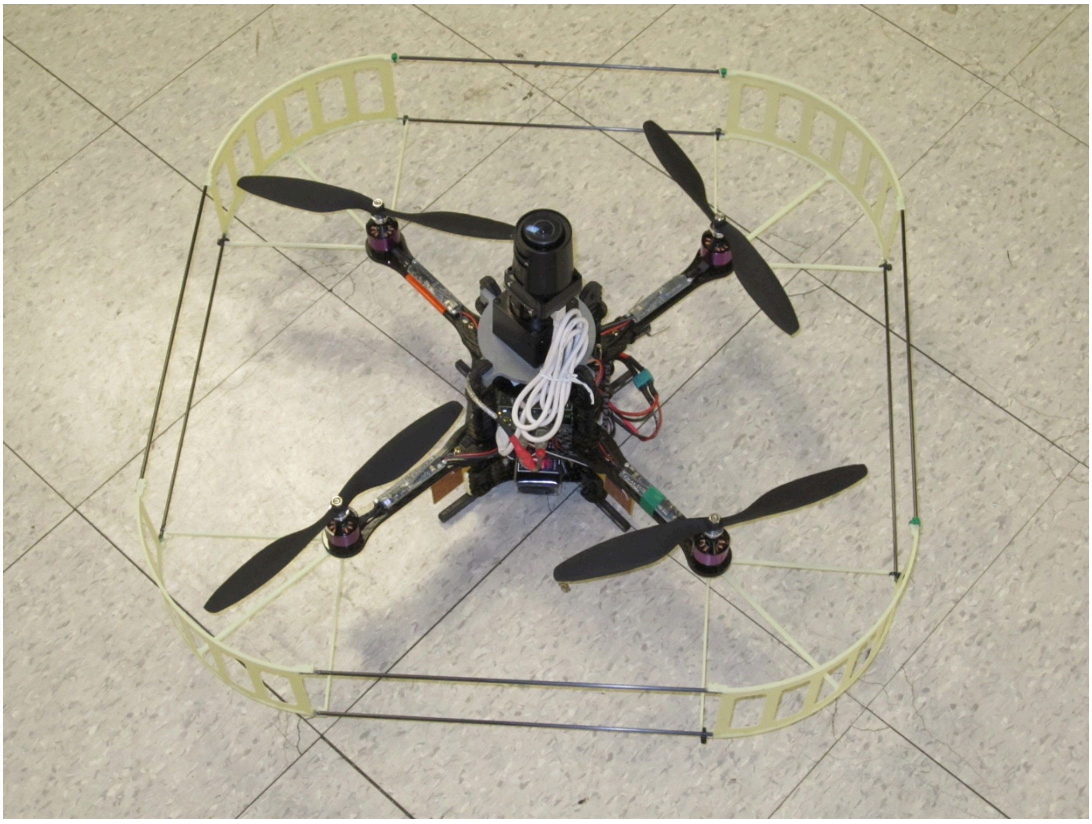

Figure 40, A helipad with an omnidirectional camera.

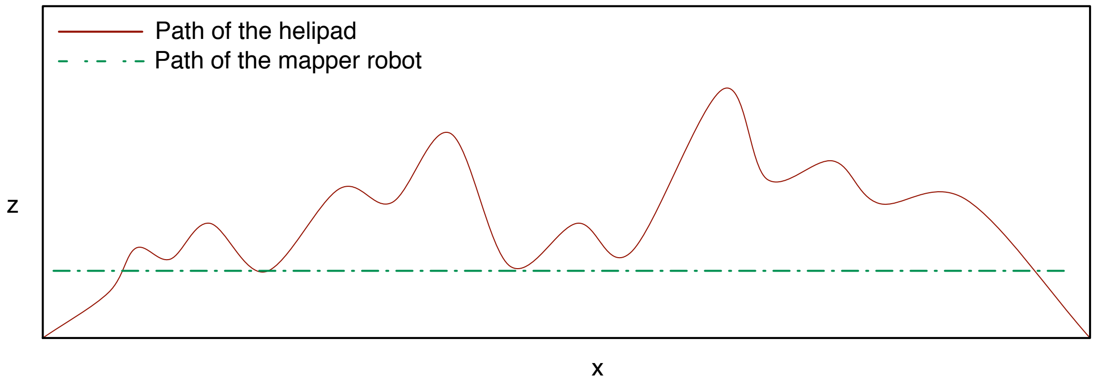

Figure 41, Approximate altitute of the helipad during the experiment

It is worthy to note that the altitude of the helipad fluctuates significantly over the course of the experiment. As illustrated in Figure 41, the helipad starts from the ground, and it gets very close to the ceiling at a few instances during the experiment. The topo-metric appearance map is generated by a wheeled mobile robot (Figure 38, left), which has a fixed camera altitude (approximately 35 cm from the ground). Despite the substantial field-of-view changes (due to the varying altitude) the helipad successfully localized itself on the map. The experiment, and the results can be seen at http://bit.ly/helipadLocalization.

Chapter 5

Conclusion

Healthcare system is the most vital foundation of a welfare society. Trends in politics, economics, demographics and technology clearly point that service robots, sooner or later, will be an indispensible part of the hospitals. Subsequently, automation is the key for maintaining the quality and sustainability of healthcare.

This thesis summarizes the efforts to address some key technical problems that concern the use of service robot in hospitals. The particular focus has been around the issues of reliability, adaptability and scalability.

In conformity with problem analysis, the current state of the logistics was studied. This study gives a good snapshot of the scale and the boundaries of the situation, which is not only valuable in the context of this thesis, but in a much wider aspect.

Obtaining the overview of the problem enables us to focus on further matters, putting the robot navigation problem to the focus. It was soon realized that sensor fusion is an important concept, and machine vision is a vital tool.

In strong compliance with the ultimate focus of the project, visual tags were investigated for their prospective use in service robot applications. Their simplicity leads to reliable solutions; they are easy to encode or decode, to install or to modify. As exemplified in multiple papers, they have versatile characteristics, which can be exploited to physically embed information into the environment and do pose estimation based on a single snapshot. These features are important to achieve adaptable systems.

Next, we proposed a fundamental framework for the navigation of service robots, using hybrid maps. The framework was developed with an overall system design in mind, and it addresses the key issues of the navigation system. Utilizing hybrid maps, the complexity of the problem is reduced and a scalable and adaptable system is achieved. Moreover, using the concept of points-of-interest and visual tags, a redundant localization system is made possible. Having a redundant layer for navigation greatly improved reliability of the system.

Consequently, the issue of annotation is addressed. It is a fundamental problem in environmental representation; which also makes the robotic systems usable by humans. We proposed a new method to physically annotate the maps and the environment, which greatly reduces the manual labor that is normally necessary; and which makes the existing annotations easily modifiable. This is an essential requirement in service robot applications, and we have proven that the representation achieved by this method is not only practical for robots, but it can also be used for human navigation.

Finally, a critical issue is resolved with a pioneering method: mapping and navigation in multi-floor buildings. The vast majority of the targeted environments for service robots are multi-floor buildings, and we presented a very simple and effective method to represent floors in terms of barometric pressures. The method is based on fusing barometric pressure data with that of range sensors; which makes it easily extendable to the most existing robot platforms.

Many of these works have already been extended, yet there is room for improvement in the future.

A natural augmentation would be studying multi-robot scenarios, where the focus shifts to optimization problems and operational research. A true automation of the logistics in a hospital is only possible by approaching the problem from system level, and based on the demonstrated capabilities of robots; the whole logistics system should be reorganized.

Similarly, another critical issue that was not touched in detail in this thesis is human-robot interaction. If the robots will operate in daily environments, they will have to cooperate or collaborate with humans. Human-robot interaction is an emerging research field, which primarily concerns with safe, reliable and efficient operation of robots. The hospital domain adds an extra significance to the necessity of studying this interaction.

As stated above, this thesis summarizes the efforts to address some key technical problems that concern the use of service robot in hospitals. A successful utilization of robots will require a much bigger effort and many more experts in a wide range of specialties. The future is inevitable, and we can only hope that this work will be a step-stone to achieve the future.

Chapter 6

Included Papers

This chapter is composed of the papers that were written in the course of the PhD project. These papers are re-formatted to match the general style and their references are consolidated at the end of the thesis.

Article I. "Service Robots for Hospitals: A Case Study of Transportation Tasks in a Hospital" Ozkil, A.G., Fan, Z., Kristensen, J.K., Dawids, S., Christensen, K.H., Aanæs H. In Proceedings of *IEEE International Conference on Automation and Logistics,* Shenyang,China, 2009.

Article II. Mobile Robot Navigation Using Visual Tags: A Review , Ozkil, A.G., Fan, Z., Kristensen, J.K., Dawids, S., Christensen, K.H., Aanæs H. In Proceedings of IASTED International Conference on Robotics and Automation, Cambridge, MA, USA, 2009.

Article III. Practical Indoor Mobile Robot Navigation Using Hybrid Maps, Ozkil, A.G., Fan, Z., Kristensen, J.K., Dawids, S., Christensen, K.H., Aanæs H. In Proceedings of IEEE/ASME International Conference on Mechatronics, Istanbul, Turkey, 2011

Article IV. Automatically Annotated Mapping of Indoor Environments Ozkil, A.G., Fan, Z., Kristensen, J.K., Dawids, S., Christensen, K.H., Aanæs H.Submitted to IEEE International Conference on Intelligent Robots and Systems, San Francisco, USA, 2011

Article V. Mapping of Multi-Floor Buildings: A Barometric Approach Ozkil, A.G., Fan, Z., Kristensen, J.K., Dawids, S., Christensen, K.H., Aanæs H. Submitted to IEEE International Conference on Intelligent Robots and Systems, San Francisco, USA, 2011

6.1 Service Robots for Hospitals: A Case Study of Transportation Tasks in a Hospital

Abstract – In this paper, the need for automated transportation systems for hospitals is investigated. Among other alternatives, mobile robots stand out as the most prominent means of automation of transportation tasks in hospitals.

Existing transportation routines of a hospital are analyzed in order to verify the need for automation and identify possible areas of improvement. The analysis shows that most of the existing transportation is carried out manually, and hospitals can greatly benefit from automated transportation. Based on the results of the analysis, three alternatives are derived for implementing mobile service robots for transportation tasks in hospitals.

6.1.1 Introduction

In this paper, it is aimed to analyze the transportation routines of a hospital, in order to understand needs for designing mobile robotic transportation systems and identify possible types of transportation tasks that can be improved by automation. As a result of the analysis, three strategies are developed and discussed for using autonomous mobile robots at hospitals for transportation: Adapting to existing transportation system, partial reconfiguration of transportation system and reconstruction of the whole system.

The paper is organized as follows. In the second section, the need for transportation automation in hospitals is elaborated and existing automation systems are presented. In the third section, Bispebjerg Hospital, where the analyses of the transportation tasks made upon is described in detail. In the fourth section, findings of the paper are discussed, and in the last section, conclusions are presented.

6.1.2 Background

6.1.2.1 A. Need for automation in hospital logistics

Hospitals are one of the vital organs of the modern society. Demographical factors, such as increasing elderly population, bring up the need for reviewing healthcare services to improve hospital operations and their efficiency and effectiveness.

The primary service provided by a hospital is patient care. In order to able to supply this service, there are a number of support services that needs to be taken care of in the hospital. Even though most of these services are invisible to patients, they have a considerable effect on how patients experience their visit to the hospital.

One of the major and usually underestimated support services in hospitals is the logistics. Main task of a hospital logistics service is to organize and maintain the material flow in the hospital. A vast variety of materials are essentially needed in a hospital, which often result in complex transportation systems and tangled flow of materials.

There is a growing interest in automation of logistics in hospitals. The main motivation behind is to reduce operational costs; nearly %30 of hospital expenses is resulting from logistics activities [123]. Statistics also show that the number of hospital patients is increasing in western countries such as Denmark [124], [125], which substantially increase the load of supporting logistic functions. Finally, variety of materials and equipment that are being used in hospitals is expanding. New supplies and equipments are being developed increasingly, and one-time use, disposable items are being more common. As a result, the volume of transportation expands substantially; more materials need to be transported more frequently.

Transportation capacity can also be improved by automation. Manual hospital transportation tasks are usually bounded with the limitations of available human work power. Routes, volumes, weights and frequencies are planned according to available means of transportation framework and personnel. An automated system, on the other hand, can yield to a much flexible transportation plan. Routes can be optimized, and more frequent deliveries can be planned for both night and day time. The speed of response to inquiries can also be improved if tracking and inventory systems and transportation system is tightly integrated.

Finally, effectiveness of everyday processes in hospitals can be improved. Staff can save considerable amounts of times, which in turn can be allocated to more patient related tasks. Therefore the quality of the service provided to the patient can be improved.

6.1.2.2 Review of the automation systems used in hospital

From certain aspects, hospitals are similar to manufacturing facilities; therefore many automation applications are adapted from these environments.

Pneumatic tube systems are one of the most commonly used delivery systems in hospitals. They consist of a tube network that links different units in the hospitals. The cargo is placed in special containers, fed to the tube system at stations and moved by pneumatic forces created by the fans and pumps connected to tube network. It is usually used for small sized cargo; such as papers, specimens or pharmaceuticals.

Track and conveyor systems are also common in hospitals for delivering relatively larger loads. Rails or conveyor belts are installed in horizontal or vertical configurations and materials are transported between floors and hospital units using self-contained lifting and dragging systems.

An alternative to such systems, which is also the focus of this study, are mobile robots. Using robotic vehicles for transportation has been of a great interest in manufacturing sites for decades, where the environment often needed to be modified. In the last decade, these systems also became popular in hospitals as localization techniques are more advanced, and range sensors are faster and more precise; which makes it possible to implement such systems with minor modifications in environments.

Helpmate was one of the pioneer implementations of automated robotic hospital transportation systems [13]. It was designed for transporting small sized cargo between departments, and it was able to autonomously navigate inside the hospitals, take elevators and avoid obstacles. Care-o-bot was developed to provide physical support to people requiring mobility and tele-presence capabilities for remote inspection, in addition to navigation capabilities [126]. I-Merc was introduced in [127], as a new and automated method of distributing meals in hospitals. In [128], a system that consists of mobile robots and automated pick-up/delivery stations is introduced. Automated stations enabled operation without the need of human loading and unloading at delivery points.

Commercial applications have also started to emerge in recent years. TUG [2-5] robot commercialized by Aethon, is a robotic system consisted of a tugging vehicle and specially designed charts to which the robot can be connected to. The system can navigate autonomously, and it also provides a system for tracking cargo using RFID technology. Speciminder [4] is another commercial robot, which is specifically designed for distributing lab specimens throughout the facilities. It has similar autonomous navigation capabilities to TUG, but the cargo container is directly placed on top of the robot. Transcar [5] developed by Swisslog is also a similar system, which consist of a laser guided low-height vehicle that can go under wheel supported cargo carts (with loads up to 450 kg), lift them up, and transport to designations that are written to the RFID tags on the carts. Lastly, FMC Technologies developed ATLIS[3], based on their AGV platform that was already being used in manufacturing plants. It is very similar to Transcar, but it is larger and has greater payload (680 kg).

It is notable that most of the systems above are successors or derivations of applications existed in other domains.

Figure 42, Map of the Bispebjerg Hospital

6.1.3 Analysis of transportations in a hospital

In this section presents the analysis of the current transportation situation in a hospital, and it is believed that a careful investigation of the transportation routines is needed to be able to design efficient robots and achieve efficiencies in a more global scale.

6.1.3.1 Description of the Bispebjerg Hospital

Bispebjerg Hospital is located in the Northwest quarter of Copenhagen, Denmark. Due to its size and architecture, it represents a typical mid-sized European hospital. It serves to a population of 255.000 inhabitants [127], with around 500 available beds and 3000 staff.

Bispebjerg Hospital was built in 1913, where different units were physically separated in a campus like architecture. An underground tunnel network connects most of the buildings. (Figure 42)

6.1.3.2 Description of internal transportation tasks:

The transportation tasks inside the hospital are classified according to the originated sources of the supplies. An overall view of the transportation tasks is given in table 1.

The tasks are described as follows:

6.1.3.2.1 Pharmaceutical supplies

Each hospital unit has a room for stocking regularly used medicines. A pharmacist is associated to a number of medicine storage rooms, whom each day checks the status of supply and orders new medicine from the central pharmacy. Unit nurses can inquire for extra medicine, using an order book. The pharmacist has the authority to register and send these orders.

Table 1, Summary of transportation tasks

Task	*Personnel involved*	*Hours/Week*
Medicine Transport	Porter (2)	50,4
Mail Delivery	Hospital Employee (4)	107,1
Transports to Unit Storages	Porter (1)	27
Transports to Sterile Cabinets	Hospital Employee (2)	69
Food Transport	Porter (5)	98
Transports from central supply	Porter (2)	51,8
Empty beds	Porter (3-4)	113,4
Transport of clothes	Porter (4)	130
Waste Collection	Porter (4)	195,84

There is a central pharmaceutical storage outside the hospital, which serves to a number of hospitals in the region. But inside the hospital, the system is distributed, so there is not a central unit for storing pharmaceuticals. Each day, the ordered medicine from the central storage is sent to the hospital in sealed boxes. These boxes are received by two porters, and then the medicine is distributed to various hospital units. At units, sealed packages are delivered in exchange of the signature of the responsible pharmacist or the head nurse of the unit. Only the pharmacist has the right to break the seal and add the medicines to the unit's inventory.

Figure 43, Carts and sealed boxes for delivering pharmaceutical supplies

Pharmaceutical waste is consisted of leftover or expired medicine. These medicines are separated and returned back to the central pharmaceutical storage in the same way new medicine arrives. Patient medicine is regarded as clinical waste and disposed into yellow contaminated waste boxes, in compliance with bio-hazardous waste disposal regulations.

6.1.3.2.2 Internal post office

The hospital has an internal post office located close to the administrative units. The internal post office is mainly responsible from distribution of external mail received from the national postal service, collection and distribution of internal mail, and collection and dissemination of mail going out of the hospital. Additionally, the internal post office is the support unit for miscellaneous urgent delivery tasks, such as transportation of medical records, test samples or results, or blood units.

There are five specified routes for mail delivery from the post office to hospital units. in each unit, the postal worker delivers the new mail and collects the mail from the outgoing box, and brings it back to the post office. Collected mail is sorted at the post office together with the external received mail, and the same routine is repeated two or three times a day, depending on the mail traffic. It is noteworthy that the mail delivery to the administrative units is done only once per day, because it is very time consuming due to the number of post boxes to visit (approximately %25 of the whole distribution routine).

The general tasks of the internal post office are handled by nine postal workers. In each round of mail delivery, four postal workers are active. It takes around 1 hour and 15 minutes to complete a round and return back to the post office. For the urgent delivery tasks, at least one person is kept on stand-by at the post office.

6.1.3.2.3 Sterile production

At the hospital, there are a total of 72 sterile cabinets in various units. These cabinets are used for keeping various sterile tools needed for the daily work. to make sure that the cabinets are equipped with the necessary content, the hospital has a department responsible for maintaining the stock and refilling of these cabinets. The department is also responsible from collecting sterilizing and packing the reusable equipment.

Everyday, each of these cabinets are visited by two porters and their content are recorded. In second round same employees revisit the cabinets where new supplies are needed. Depending on the unit, each cabinet is refilled one to three times a week.

Every morning, a porter collects used tools. Porter visits hospital's units, collects the tools in a large cart, and returns them back to sterile production unit for cleaning and re-sterilizing.

6.1.3.2.4 Central sterile products storage

The sterile products storage is responsible from storing both sterile and non-sterile single use products.

Ordering of non-sterile products is done through an internal online ordering system automatically, so that the storage keeps a minimum amount of supplies. Single use products are handled in the same way as sterile tools coming from sterile production unit, and are kept in the same sterile cabinets. These products are delivered together with the sterile tools, and then disposed to contaminated waste boxes.

6.1.3.2.5 Kitchen

Each hospital unit order food needed using the same ordering system as sterile products storage. The orders are processed in the hospital kitchen and food is distributed to units in two rounds: at 7 AM for the breakfast and at 11:30 AM for lunch and dinner. Food is delivered to units in special carts with heating and refrigeration (Figure 44). Each unit is responsible from the internal distribution of the food and supplement of tableware. One hour after the delivery, the empty carts are collected by the same porter that makes the delivery. Apart from meals, kitchen also supplies groceries such as bread, snacks and drinks to units. Each unit consumes a cart of groceries per week, and everyday, six to nine carts of groceries are distributed. Bottles are collected from hospital units and returned to the kitchen storage.

Figure 44, Food cart with heating and cooling

6.1.3.2.6 Central storage

Central storage is the main storage facility of the hospital. Currently 1296 stock articles are handled by the central storage.

Hospital units place their orders weekly, using an online ordering system and receive their orders on a specific day of the week. Three carts are available for transportation of good from the central storage. There are rounds throughout the day, and in each round there are 3 to 5 stops. Goods are only going out of the central storage; therefore each cart comes back empty at the end of the round.

6.1.3.2.7 Bed maintenance

Each time a bed patient is discharged, the used bed is sent to the bed maintenance central for cleaning. During the first half of 2008, an average of 2700 beds were processed per month.

Used beds are usually collected by the department's three porters, or sometimes sent directly from the unit the patient is discharged. When a bed arrives, sheet, cover, quilt and pillow is removed and delivered to the laundry central. Each bed is identified with a barcode, and based on the history of its circulation; the bed undergoes a certain cleaning routine. Cleaned beds are remade using clean linens, and kept in a storage zone, as seen in Figure 45, which can store up to 80 beds.

6.1.3.2.8 Laundry central

Laundry central in the Bispebjerg hospital serves to a number of hospitals in the region. In total, % 20 of the production is allocated to Bispebjerg.

Everyday, 30 -45 carts of clean clothes are distributed, whereas 25-30 carts of dirty clothes are collected in the hospital. Personnel uniforms are handled separately from the other cloths, where the staff can pick up clean uniforms from special cloth boxes.

One unit employee goes rounds at 60 different storage rooms of the hospital and estimates the needed supplies. Based on these estimations, clean cloths are delivered to these rooms and dirty ones are collected subsequently. Dirty cloths weigh %10-12 higher than the clean cloths per unit cargo.

Figure 45, Clean beds are kept in the buffer zone

6.1.3.2.9 Refuse disposal / renovation

Refuse disposal unit is responsible from gathering and sorting hospital waste. There are several types of waste being collected. These types and corresponding amounts of waste collected in 2007 are given in table 2.

There are certain regulations on disposal of waste in hospitals, as mentioned in the pharmaceutical supplies section. Each hospital unit is responsible from sorting their own waste. Clinical waste must be kept in sealed containers in disposal rooms that are located in each floor or unit of the hospital. Most of the waste generated in the hospital is stored in disposal rooms, but some of the general-purpose waste is disposed to garbage chutes.

Table 3, Types and amounts of waste collected in 2007

Category	*Weight (in tonnes)*
Refuse	813,78
Clinical waste	46,64
Cardboard (recycled)	48,14
Paper (recycled)	45,65
Bulky refuse	82,52
Construction waste	28,12
Glass	7,37
Electronics	3,21
Refrigerants	12
Iron and metal	10,86

6.1.4 Discussion

The analysis clearly shows that the transportation system in the hospital is very complex. Some of the sources of the complexity are identified as the different sources of material flow, different time constraints on distribution routines, work regulations of the personnel and transportation equipment being used.

Based on the findings, three alternatives are devised for the implementation of autonomous robots in transportation systems in hospitals:

4. Adaptation to the existing system: The transportation system remains the same in the global level. In certain types of transportations, robots are used to fulfil existing transporting tasks. Routines remain as they are.

5. Partial reconfiguration of the system: Certain tasks are identified, and robots for these tasks are designed for optimal performance. Transportation routines are

optimized based on the capabilities of robots. Existing storage facilities are utilized, and the delivery routines are stretched to 24-hour operation.

6. Restructuring of the system: In order to facilitate system wide optimization, a central stock system is implemented. Various types of materials can be stored in the central stock. The main advantage of this system is that it is possible to combine different types of materials at the central stock and send them together; so that number of dispatches is minimized. Different types of robots are introduced accordingly, and routes are optimized for 24-hour operation.

It is possible to say that most of the robot systems presented in the previous section fall into the first alternative. It is the easiest option in terms of implementation overhead; since there are minor alterations on the routines and the robot transporters are simply take place of the human counterparts wherever feasible.

Service robots in hospitals are still not very common and existing ones are under active development. Therefore it is normal to observe that they simply try to adapt to existing transportation systems. But it can also be claimed that in this alternative, the potential of using autonomous mobile robots and automation is not fully exploited.

In order to utilize the most out of an automated robotic transportation system, at least a partial reconfiguration or reconstruction is needed. It is obvious that in current case, several units are needed to be visited multiple times by different porters. Instead, supplies from sterile central, sterile production, central storage, pharmacy and the central laundry can be combined and delivered together in a single cargo unit. The incoming cargo then can be sorted out and distributed within the unit. In the same way outgoing materials can be assembled at the unit and placed in a single cargo unit. It is also possible to operate day and night to make several collections and distributions.

It is believed that, even though it is more difficult to implement compared to the first one, the second alternative is much better at taking the advantage of using mobile robots for transportation. In case of going into renovations or restructuring, existing hospitals should consider this option instead of the first one.

The third alternative is the most optimal way of designing an automated transportation system for hospitals. The aim is to restructure the whole system, and create a logistics center instead of various decentralized storage units. In this alternative, even more emphasis can be put on 24 hour operation.

Due to the need for substantial amount of environmental modifications this alternative would require, it is not a feasible option for existing hospitals. But while new hospitals are designed, it should be the first option to consider.

The analysis also revealed that such an automated system has the potential to decrease the risk of infections. Equipment being used for transportations as well as commonly touched surfaces such as doors and elevators are sensitive places in terms of infection risk. An easy to clean robot system, as well as automated door and elevators needed for the robot operation can improve the situation.

Lastly, it should be remarked that the integration of IT technologies for ordering of supplies and planning of resources, into the transportation systems utilizing robots is another important issue that can boost the efficiency and the effectiveness. There is still a lot of room for improvement in this integration, and much progress is expected in this field.

6.1.5 Conclusion

This paper presents the need for automated transportation systems for hospitals, gives an analysis of the existing transportation tasks based on a case study, and outlines alternative scenarios for implementation.

It is found that need for mobile robots in hospital transportation tasks is evident, but their deployment needs to be based on a thorough analysis of the whole logistics system, and their designs should be made accordingly.

While existing robotic systems show the potential of future applications, significant improvements can be made by fully or partially reconstructing the transportation frameworks.

6.2 Mobile Robot Navigation Using Visual Tags: A Review

Abstract-This paper aims to give an overview of visual tagging systems and review robotics applications that use visual tags as navigational aids. Two main classes of visual tags are barcodes and augmented reality markers. Barcodes are very commonly used in various industrial applications, and they exist in one- or two-dimensional form. Augmented reality markers have relatively limited use, and they are classified as template based markers, id based markers and topology based markers. A number of robotic applications have already utilized visual tags to solve the navigation problem. In most of these applications, a new tagging system is developed and used in the limited context of the application.

As a result of the review of these visual tagging systems it is concluded that augmented reality markers are the most suitable visual tags that can be used in navigational tasks of mobile robots.

6.2.1 Introduction

A visual tag can be defined as a graphical identifier which is easy to distinguish and recognize in the environment it is placed. Because of the apparent need for such a system, a number of visual tags have been developed and are being used in a big variety of applications, including robotics.

This paper addresses three relevant issues. First, (non-robotic) visual tagging systems and their applications are reviewed. Secondly, visual tagging systems in mobile robot navigation tasks are investigated. And finally these systems are compared and suggestions for the use of visual tagging systems in mobile robot navigation tasks are presented.

The rest of the paper is organized as follows. In the second section, visual tagging systems that are used in a range of non-robotic applications are reviewed and classified. In the third section, visual tagging systems that are used or developed in robotic applications are investigated. In the fourth section, reviewed systems are compared and their (possible) uses in mobile robot navigation are discussed. Finally, the paper is concluded and suggestions for future applications are presented.

6.2.2 Visual Tagging Systems in General

Visual tagging systems are being used in a wide area of applications that are not directly related to robotics. These systems can be reviewed under two main categories: Barcodes and augmented reality markers. Barcodes can be further divided into one or two-dimensional barcodes, and augmented reality markers can be further divided into template based, id based or topology based markers. The rest of this section explains them in more details.

6.2.2.1 Barcodes

Barcodes are machine-readable representation of data [129], where data is encoded using simple graphical elements and decoded optically using digital imaging systems.

There are two types of barcodes: One and two dimensional barcodes. One dimensional barcodes decode data using bars with different widths and contrasts. Two dimensional barcodes use dots, rectangles or hexagons, and these graphical elements are spread to two axes. Following sections outline commonly used one and two dimensional barcodes.

6.2.2.1.1 1-D Barcodes

6.2.2.1.1.1 UPC /EAN Barcode

UPC/EAN barcode is a numbering system, which has a fixed length of 13 digits (Figure 46-a). Each digit is represented by a group of black and white bars. Digits of the UPC/EAN barcode are divided into four parts as follows. First two digits is called as the number system, which encodes the country the barcode is issued. Next five digits specify the manufacturer code, which is issued by the numbering authority to a specific manufacturer. The following five digits are reserved as the product code, which is assigned by the manufacturer. Finally the last digit is used as a parity check digit.

Figure 46, examples of one dimensional barcodes: (a) EAN, (b) Code 128, (c) Postnet / Royalmail

6.2.2.1.1.2 Code 128

Code 128 can encode 128 characters of the ASCII character set. In contrast to UPC/EAN barcode, Code 128 is extendable: the barcode can be as long as needed. In that sense, Code 128 and its derivatives can be regarded as a special, machine-readable alphabet. Each letter of this 'alphabet' is composed of six bars with varying widths (Figure 46-b).

6.2.2.1.1.3 Postnet Code

Postal Numeric Encoding Technique (Postnet) is developed to automatically sort mail by encoding and decoding ZIP/postal codes [130]. Postnet codes encode numeric data, and they consist of bars with varying lengths.

6.2.2.1.2 2-D Barcodes

6.2.2.1.2.1 QR Code

Quick Response (QR) code is a two-dimensional matrix symbology that use square for encoding. It was designed by Denso-Wave in 1994 for tracking parts in vehicle manufacturing [131], [132].

Wide use of QR codes started in Japan, as mobile phone manufacturers started to ship camera phones with built-in QR Code decoding capabilities. Since then, QR Codes became a common tool for marketing and information exchange. It has been used in a range of media, such as newspapers and magazines, billboards, business cards, ticketing systems and public transportation systems.

QR Codes can be identified by three positioning marks on the top left, top right and bottom left corners (Figure 47-a). Data is decoded by locating these corners on an image and examining the area specified by these corners.

QR Codes implement Reed-Solomon error correction [133]. Data is encoded redundantly and this makes it possible to decode partially damaged or occluded barcodes. It is possible to encode alphanumerical data with variable size.

6.2.2.1.2.2 Datamatrix

A Datamatrix encode information inside a rectangular area that is defined by a positioning pattern (solid edges on the left and bottom), and a timing pattern (right and top) (Figure 47-b) [134]. Data is encoded as a matrix of contrasting squares inside this area. Similar to QR Codes, Datamatrix use error checking and correction and they can encode alphanumerical data.

6.2.2.1.2.3 Shotcode

Shotcode is a circular barcode that can encode numeric data (Figure 47-c). It was originally developed as a low cost method to track locations [135].

Main feature of Shotcode is the representation of data in polar coordinates. Resulting image has a dartboard look, with a solid outer edge for area estimation and a bull's-eye pattern at the center for positioning.

6.2.2.1.2.4 High Capacity Color Barcode

High Capacity Color Barcode, (HCCB) is a matrix type barcode symbology, which uses color as an extra dimension [136]. It is developed by Microsoft for mobile marketing and product identification.

HCCB barcode consists of a solid dark frame and a grid of colored data nodes from a set of four or eight colors. Due to its third dimension, HCCB barcode result in smaller footprints than other barcode types. HCCB is a proprietary system, and it decodes the data as a pointer to actual data that is stored in a remote server.

6.2.2.2 *Augmented Reality Markers*

Augmented reality is a term to define interactive applications, in which 3-D virtual objects are integrated into 3-D real environment imagery in real time [137]. Therefore, an important part of an augmented reality application is to extract position information from the video input.

In augmented reality, 3-D registration of a scene from an image is usually accomplished by using markers - special patterns that are easy to distinguish in respective environments. Several markerless augmented reality applications have recently emerged [138-142], but it is anticipated that marker based systems will remain dominant [143] because of they are much simpler to use and require relatively low resources.

A marker based augmented reality application typically works as follows (Figure 48). Upon acquisition of and binarization of a frame from video source, the characteristic shape of the marker is looked for in the image. If the physical characteristics of the marker are known and the camera is calibrated, the actual and the perceived shape of the marker are correlated and the pose of the maker (relative to the camera) is estimated. Next, a specific pattern is searched inside the boundary of the marker image, and matched against the database of known markers. Finally, the corresponding virtual object is instantiated and overlaid on the video of the real scene.

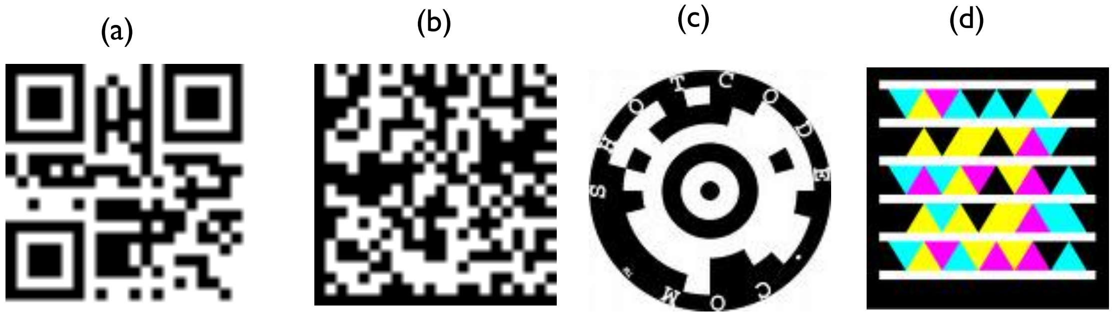

Figure 47, examples of two dimensional barcodes: (a) QR code, (b) Datamatrix, (c) Shotcode, (d)HCCB

Marker based augmented reality have been an active field of research during the last decade, which resulted in a number of marker designs. These markers are categorized in three groups as Template based markers, Id based markers and

Topological markers (Figure 49). They are described in detail in the following sections.

6.2.2.2.1 Template Based Markers

Template based markers were introduced in ARToolKit, an open source augmented reality software library [144]. The ARToolKit framework consists of: (1) a marker system, (2) a marker positioning and tracking module, and (3) a 3d graphics handling module for 3d object generation and video manipulation. They are usually composed of a relatively thick, black frame and a monochrome pictogram inside.

The pictogram can be anything; a number, a letter or a simple picture, but complex shapes will degrade identification and tracking [145]. Furthermore, the system must be trained beforehand to register the markers to a database.

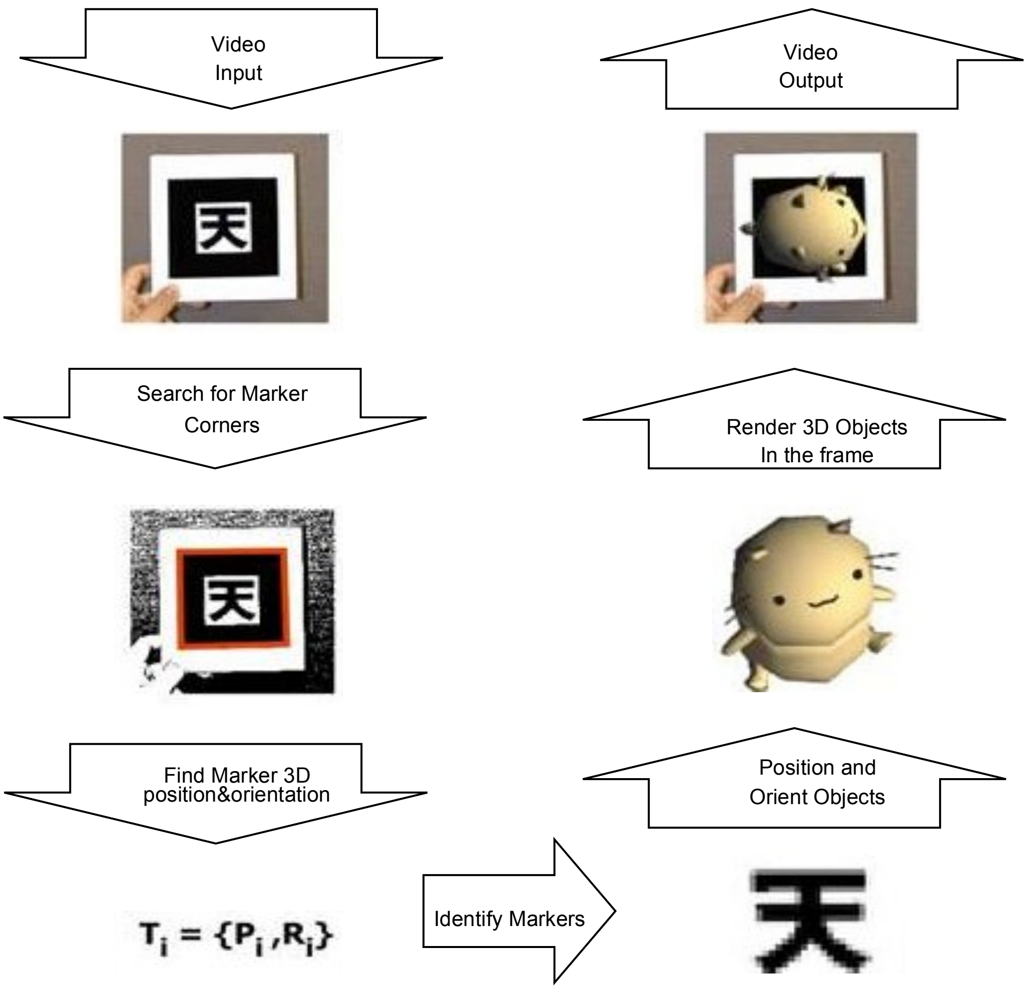

Figure 48, Workflow of ARToolKit [144]

6.2.2.2.2 Id Based Markers

Id based markers visually encode numerical data on the marker, similar to 2-D barcodes. A certain area on the marker is represented as a grid of contrasting cells, where each cell can store one bit data. Data capacity, therefore the number of different markers is limited by the number of grid cells and the size of the marker.

Markers could appear from any viewpoint in augmented reality applications. In order to facilitate rotationally invariant marker detection, certain cells in the grid are reserved to indicate the orientation, which can be utilized in robot applications. Similarly, certain grid configurations are discarded due to the effect of symmetry.

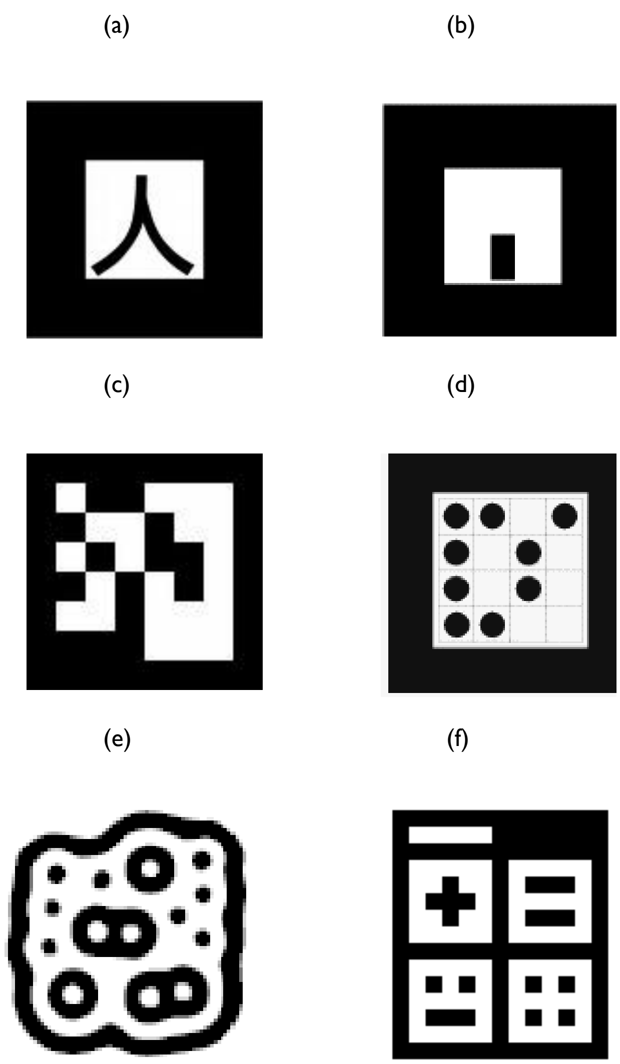

Figure 49, Examples of augmented reality markers: (a,b) Template based [144], (c,d) Id based[110], [110], (e,f) Topology based [146], [147]

6.2.2.2.3 Topology Based Markers

A topology-based marker is identified by the topology tree of its graphical elements and their connectivity (Figure 50). Since there are no geometrical constraints, they can be designed in very different shapes and yet represent the same topology tree.

The main advantage of the topology-based markers is their robustness, due to the fact that there is no shape matching during identification process. Therefore, identification of topology based markers is less sensitive to the effects of noise, blur, affine/perspective transformations or warping.

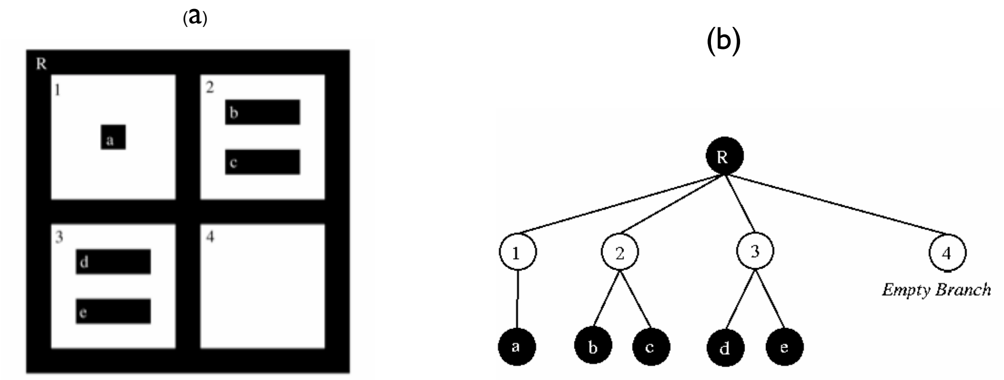

Figure 50, A topological marker. Marker (a) is represented by its connected components (b). (Taken from [147])

6.2.3 Visual Tagging Systems in Mobile Robotics

Machine vision has been more and more included in autonomous robotics research during last decade [148]. Several problems in robotics, such as mapping, localization, object manipulation and obstacle avoidance rely on finding features/landmarks in images, where the definition of a landmark varies depending on the context of the application.

A landmark can be natural or artificial, based on how they appear on the scene. Natural landmarks occur in the scenes natively, and they are generally identified by local image feature detectors and descriptors, such as SIFT, SURF or MSER. (Please refer to [149] for an extensive survey of local image features). On the other hand, artificial landmarks are designed according to the characteristics of the environment they are going to be placed and the detection methods to be used. An artificial landmark is, simply, a visual tag; and the rest of this section outlines a review of artificial landmarks used in mobile robotics.

6.2.3.1 Review of Artificial Landmarks

Kabuka and Arenas designed a standard pattern for position verification of a mobile robot in [150]. It consists of black stripes placed to the both sides of a circle, where the circle is black in one half and white on the other. Based on the geometric properties of the projection of a circle in the image plane of a camera, they developed an algorithm to estimate the distance, elevation and co-azimuth angles of the viewing point. The main drawback of this implementation is that the camera has to be at the same elevation with landmarks.

In [151], Becker et. al. used artificial landmarks for mobile robot localization in an indoor workspace. Their landmarks act as the "distinctive features" at predefined positions in the environment. The landmark consisted of three elements, an inner 3x3 grid of black and white cells, an outer circle and an opening on this circle. The detection method was tested in two scenarios. In the first one, the landmarks were placed on the ceiling, where the robot had an upward-looking camera. In the second, a forward-looking stereo camera setup was used and the landmarks were mounted at the eyesight on the walls. It is reported that ceiling landmarks result in better accuracy for localization purposes, due to the fact that the image of a landmark is directly related to the position of the robot in this setup. It is also more reliable to detect this kind of landmarks because of the low chance of obstruction of the camera view. On the other hand, it is easier to attach landmarks on the walls and they can be seen from relatively larger angles, but the obstructions are more likely.

Lin and Tummala described a robot navigation system in [152], where they designed a rather complicated 3D landmark for position re-calibration. Their landmark consist of a white background, a black circle on this background, and a smaller white circle, which is protruded 12 cm from the center of the black circle. Because the marker has depth; the smaller white circle appears to be on the left/right half of the black circle when it is viewed from the right/left of the landmark. Using Elliptical Hough transformation, the viewpoint of the camera is determined from the nested circles found on the image using elliptical Hough transformation.

Taylor et al. used a specially designed barcode that consisted of five grayscale strips for motion planning and exploration in [153]. They used vertical edge detection to search for six consecutive edges that define the landmark. They used 11 wall-mounted landmarks to partition a 350 m^2 office area to generate a connectivity graph. As the markers are attached to walls, they should not be occluded for this method to perform.

Tashiro et al emphasized on optimal design and arrangement of artificial landmarks for aiding robot navigation in [154]. Their context of optimization is based on the minimum number of landmarks that ensures relative pose estimation from the landmark, with a specified maximum error. For that purpose, they developed a 3D "signboard", where two identical boards are attached together at an angle of 140 degrees. 4 LEDs were also used to mark the corners of the signboard. Optimization problem were addressed in a simulated environment, where 10 landmarks were used in a 25m^2 area.

Jang et al. used a similar looking marker to that of [154] in [155]. Instead of LEDs they use a color pattern that is composed of two vertically neighboring color patches. The color pattern is positioned at 45 degrees with respect to two orthogonal supporting planes in white, and a black line is drawn where the supporting planes meet. Therefore, the position of the line on the image is relative to the orientation of the camera. They developed a probabilistic method to track the position of the landmark, based on Condensation algorithm [156]. The method was tested in a 60m2 office environment using 7 landmarks, and it was shown that tracking performance is improved. This method is more sensitive to lighting conditions, since it is based on color properties of image.

One of the most notable applications where artificial landmarks were used is the museum tour guide Sage [157]. It was used in Carnegie museum of Natural History, where it remained in service for over 9 months. A 3-D marker, similar to that of [152] was developed, where a black square is protruded from a distinctly colored background square. The landmarks were placed at well-visible positions in the museum, and they helped the robot to correct cumulative encoder errors. A 2D version of the landmark was also used in [158] for visual servoing of the robot in the vicinity of the charging station.

Scharstein and Briggs introduced "self-similar landmarks" in [159]. The landmark is designed after so-called self-similar functions, which are periodic and scale invariant. Self-similar landmarks consist of a positioning pattern and a code pattern. The positioning pattern is modeled after self-similar function, and the code pattern is an 11 bit array of cells which is attached to the end of the positioning pattern.

Self-similar landmarks were first used for navigation in [160]. An office environment was marked with several landmarks, and a mobile robot was guided through the environment for learning the connectivity map. This map was consequently used for topological localization. Detection method was further improved in [161] by imposing restriction on rotational invariance, to reduce the amount of calculations. Similarly, Pradalier used self-similar landmarks for vision based handling tasks for an autonomous outdoor forklift [162]. Finally, Celaya et al modified the landmark in [163] so that 5 bits of information can be encoded in the self-similar pattern. The modified landmarks were used for assisting the navigation task of a 6-legged outdoor robot.

Sattar et al. introduced Fourier Tags in [164], to provide reliable ground truth position estimation where viewing conditions are deteriorated due to distance, noise, fog etc. Fourier tags encode numeric data by converting their binary forms into frequency domain. Resulting graph is then revolved around the z-axis to achieve a circular tag. Tags are recognized by detecting circles in the image, sampling a line segment that passes through the center of the detected circles, and searching for encoded data in the line segment using fast Fourier transform.

Fiala developed a visual navigation system, based on an omnidirectional image sensor and navigational markers that are placed in an indoor environment in [165]. Landmarks are vertically placed 1D barcodes, which are encoded similar to code-128. A catadioptric camera is placed on the robot parallel to the ground, so that the barcodes point to the center of the image. The image is scanned from its center using polar coordinates to detect barcodes. While simulations and experiments with real imagery proved the usability, the method is not very practical because the markers are significantly large (20x200cm)

Fiala also developed a system where ARToolKit markers were used for vision-guided control of multiple robots in [166]. The system consisted of a group of small robots, each marked with a marker, a camera that has the view of the robots' rectangular workspace, and a central computer. The corners of the workspace are also marked with fiducials, which made it possible to estimate positions of robots. The method is claimed to be particularly useful in tabletop setups for swarm robotics experiments.

Orqueda and Fierro used markers that are very similar to id based augmented reality markers for relative localization of multi-robot teams in [167]. Markers are placed around the peripheral of robots, where other robots can estimate relative positions of markers and localize themselves. The main feature of this work is the use of unscented Kalman filter for improved distance/range estimation. Simulations showed that the method offers near real-time pose estimation, but noise and changes in lighting can highly degrade the performance.

Li et al. proposed a domestic service robot framework based on artificial landmarks in [168]. The service robot is a mobile robot equipped with a robotic arm, and the objects and furniture's in the environment are marked with a circular marker with a QR-code barcode at its center. These markers are used for both locating objects and gathering information about the objects through the encoded data in the QR-code. The work presented in [168] is a design proposal and no results have been achieved, but it differs from the rest of applications in the way the markers are used to provide direct information about the environment.

6.2.4 Discussion

The review clearly shows that a great variety of visual tagging systems have been developed and used in an even greater variety of applications. Based on findings, it can be argued that such a variety results from three main reasons.

First, the need for more and more automation in industry created a demand for machine readability of items. Due to variations in mass production processes and different sizes, shapes or materials of products, several systems emerged for different segments of industries. Secondly, major source of variety emerged after the visual tagging systems were introduced into public services, such as mail, tolling/ticketing systems and personal identification systems. Finally, the required technology has been more reachable as the computing devices became highly ubiquitous. Mass popularization of camera equipped mobile phones made it possible to use visual tagging systems in a new context, and a big number of 2d barcode symbologies have been developed for this market.

The most notable finding out of the survey is that nearly all of the reviewed robotics applications designed and implemented their own proprietary visual tagging system. They are not only similar to each other, but also they have similarities with 1-D barcodes [159], [165], 2-D barcodes [168], [169] or augmented reality markers [170]. Therefore, much of the basic work is repeated but not reused, resulting in incompatible systems.

Template based and topology based AR markers need to be designed and trained beforehand. Performances of these types of markers are very dependent on marker design. On the other hand, id based markers can be used without training. They have relatively low data capacity compared to 2d barcodes, usually in the order of a few thousand distinct markers. They need to be used with a lookup table, where the marker id points to relevant information that is compiled and stored beforehand.

2-D barcodes also have several advantages that can make them suitable for robotics applications. They can carry relatively more data in smaller footprints, and they are designed to be read using low-cost cameras. They incorporate error checking and correction algorithms, which makes them robust to occlusions and partial damages. Their data capacity is superior to other symbologies. They do not require high processing powers and many open-source software libraries for encoding and decoding 2d barcodes are readily available, due to the large user base. The major disadvantage of 2-D barcodes is the need for relatively controlled environments for decoding. They are not very robust to changes in viewpoints and blur, which makes them rather unsuitable for being attached to walls in robotics applications. Resolution of the camera or the size of the barcode should also be relatively proportional to the data encoded on the barcode.

Most suitable type of visual tags that can be used to robotics is augmented reality markers. They are designed to provide near real-time detection and tracking using inexpensive cameras and they are robust to viewpoint changes. Furthermore, template or id based markers can be used to estimate the relative pose of the robot. This property can be used for metric localization or visual servoing tasks.

6.2.5 Conclusion

There are many types of visual tagging systems that are being used in a great variety of applications. In this paper, an overview of existing systems is given. These systems are divided into two main classes and characteristics of their sub-classes are portrayed. Following, the uses of visual tags in robotics is reviewed. It is seen that visual tagging has been an active field of research in robotics and several applications has emerged. Finally, the findings of the review on visual tags are discussed, where existing systems have been compared in the context of robotics. Due to their similar working principles, AR markers stand out as the most suitable visual tagging type that can be adapted to robotics use.

It is concluded that visual tagging systems can be used as effective and easy tools for building robotics applications, but there is not enough exchange and interaction between the robotics community and the others that work with visual tags. Active and close collaboration between robotics researchers and especially AR researches can result in advanced and improved applications.

6.3 Practical Indoor Mobile Robot Navigation Using Hybrid Maps

Abstract—This paper presents a practical navigation scheme for indoor mobile robots using hybrid maps. The method makes use of metric maps for local navigation and a topological map for global path planning. Metric maps are generated as 2D occupancy grids by a range sensor to represent local information about partial areas. The global topological map is used to indicate the connectivity of the 'places-of-interests' in the environment and the interconnectivity of the local maps. Visual tags on the ceiling to be detected by the robot provide valuable information and contribute to reliable localization. The navigation scheme based on the hybrid metric-topological maps is scalable and adaptable since new local maps can be easily added to the global topology, and the method can be deployed with minimum amount of modification if new areas are to be explored. The method is implemented successfully on a physical robot and evaluated in a hospital environment.

6.3.1 Introduction

Map-based navigation has been an active field of research since very early days of mobile robotics, and a number of map-based methods have been introduced to solve the problem of navigation. Representation of the environment, or modeling the world, is the key feature in map-based navigation, and it depends on several factors such as sensing capabilities, processing capabilities and the environment itself.

In this paper, we propose a practical indoor mobile robot navigation method using hybrid maps. A Hybrid map, in general, combines multiple maps of same or different kind to represent the same environment.

The method uses metric-topological hybrid maps are to represent indoor environments. Metric maps, in the form of occupancy grids represents local environment accurately for localization and obstacle avoidance. The topological map abstracts the environmental representation with nodes and edges, which is useful for global path planning and symbolic problem solving. Finally, use of visual tags in topological map allows efficient means for localization in both local metric maps and in global topology map.

Rest of the paper is organized as follows. In the next section, the background of the proposed method is presented. In the third section the method is described and typical usage scenario is explained. In the fourth section, physical experiments and evaluations are given, and in the final sections, discussion of the method is depicted and concluding remarks are presented.

6.3.2 Background

Autonomous navigation of indoor mobile robots has been a popular topic both in industry and research community. Many solutions has been proposed and implemented for autonomous navigation in industrial settings, where the environment is controllable and usually significant modifications are necessary. Automatic guided vehicles (AGV) traditionally used deterministic methods for navigation, such as magnets embedded in the floor [171], tracks painted with fluorescent ink [172], reflector strips attached to the wall [173] and recently, RFID tags [174] or optical markers placed on the floor[175].

On the other hand, with the influence of deliberate control paradigms and statistical theory; several methods has been developed by the research community for map based robot navigation, which did not require extensive environmental modifications.

A number of map types are in use today for mobile robot navigation. In the order of popularity, these can be classified as; metric maps[176], topological maps [177], sensor-level maps [178], appearance-based maps [179] and semantic maps [58].

Simply put, metric maps represent the environment in terms of distances that correspond to actually distances of the objects in the real environment. Occupancy grid maps [41] are the most commonly used metric maps, where the environment is represented as a matrix of cells and each cell has a value that correspond to the probability of its occupancy.

In contrast, topology maps represents the environment in terms of its constituents and their connectivity without using metric information. Therefore, topology maps mainly describe the structure of the environment, and how places or objects (nodes) are related to each other (edges).

Metric and topological maps are alternative ways of representing environments, and they have complementary strengths and weaknesses. Metric maps are easy to build, represent and maintain for small environments [45]. They can be used for accurate localization and for optimal route planning. Metric maps also layout the environment in an easily readable way for humans [36]. Topological maps are very suitable for planning, and they are easy to scale up for large environments. Sensor precision is not as important, since there is no need for precise position estimation for map building [180]. Topological maps are also good interfaces for symbolic problem solvers.

Hybrid metric-topological maps simply combine these advantages by utilizing the appropriate type of map for different tasks. The idea of hybrid maps was already suggested in early work ([48], [58], [178]), but a number of new methods and implementations (such as [46], [52], [53], [181], [182]) are introduced in the last decade.

6.3.3 Method

The method consists of two phases; map generation and actual navigation. Map generation principally takes place offline; pre-recorded environmental data is processed together with user input. At the end of this phase, a hybrid (metric-topological) map is obtained.

During navigation phase, the map generated in the mapping phase is used for local and global navigation. Structural information about the environment in the topology map is also used for user interaction by extracting the 'places-of-interest' and prompting them to the user to select a destination.

A significant characteristic of the method is the use of visual tags to aid navigation. Visual tags are basically artificial landmarks, that are easy to deploy and easy to distinguish and detect, and they correspond to certain nodes of the topology map.

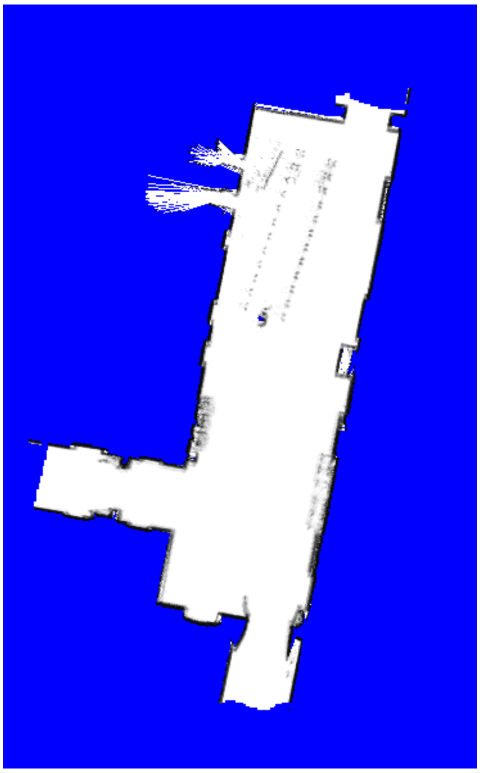

Figure 51, Occupancy grid map of Waiting room at Building 60, Bispebjerg Hospital, Denmark

6.3.3.1 Map Generation

In this phase, metric-topological map that is required for navigation is generated. The resulting hybrid map is hierarchical in the sense that; in the lower level, metric maps represent a number of regions in the environment and in the higher level, the topology map represent the interconnectivity of these regions.

Map generation is divided into 3 stages: Metric map generation, annotation and topology generation.

6.3.3.1.1 Metric map generation

Metric maps in this method are simply occupancy grids that represent static, local knowledge of certain regions in the environment.

Generation of these maps require pre-recorded datasets of these regions. Datasets are collected by manually guiding the robot in these regions. To be able to create connectivity in the topology generation step of this phase, it is assumed that a metric map partially overlaps with at least one other metric map. Therefore certain places in regions need to be visited more than once during data collection.

Datasets consist of odometry correlated range data and sequence of images that comes from onboard sensors of the robot. A number of sensor systems, such as monocular/stereo cameras, acoustic or infrared sensors can be used to record range data, but laser range finders are the most common sensors to generate occupancy grid maps, due to their superior range and accuracy.

Datasets can be collected by a single robot over the time, or by a group of robots simultaneously. The only important consideration is the partial overlap, which is significant enough do correlation. Consequently, it is also assumed that the environment is relatively static around the overlapping areas, i.e. no significant changes occur between revisits during data collection.

After obtaining a group of datasets, occupancy grid maps are generated. This is essentially a simultaneous localization and mapping process where probabilistic methods are utilized to represent the region as a grid and estimate the occupancy likelihood of each cell. Occupancy grid maps can be graphically represented where darker pixels account for high probability of occupancy.

Metric map generation is an offline process. At the end of this process, a set of metric maps (such as in Figure 51) is obtained.

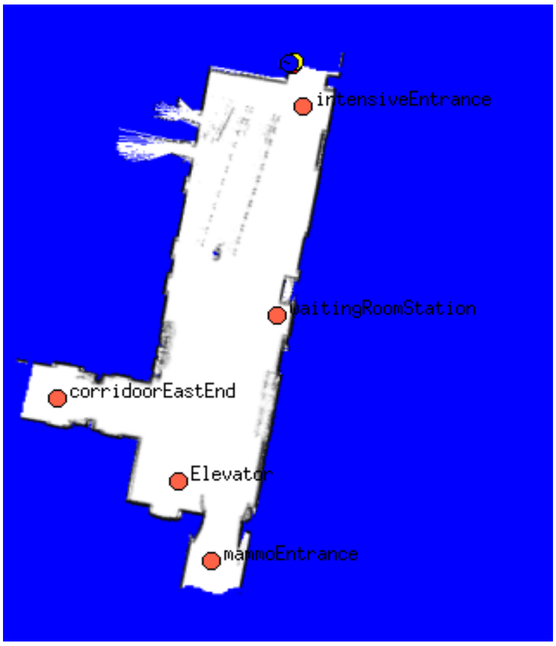

Figure 52, Annotated version of the map in Figure 1

6.3.3.1.2 Annotation

Annotation is the consequent phase to metric mapping, in which semantic information is manually added to existing metric maps. The main reason for having this phase is to address the following question: '*where* the robot should go?'. To elaborate, consider the following usage scenario of an indoor service robot; where the user would like to send a robot to a certain 'place-of-interest'. While this place-of-interest has a meaning for the human user in terms of natural language, such as 'storage room', it only correspond to a certain pixel of the grid map, or to latitude longitude and orientation of a pose in the metric map. Annotation phase merely creates a link between semantic knowledge about the environment and its representation in robot coordinates.

During annotation, two types of information are added to metric maps; actual places-of-interests, i.e. names of the places that can be reached by the robot and 'switching nodes' where two or more maps overlap. These overlaps are also marked with distinct visual tags, which act as artificial landmarks. Both types of additions correspond to a certain position in the map, whereas switching nodes additionally refer to the distinct ID of the visual tag at that node.

Implicitly, an annotated map also depicts local topology (Figure 53). As all the nodes of a local topology map reside in the same metric map, any local node can be reached from any other local node. At the end of this phase, a set of annotated metric maps (such as in Figure 52) are obtained.

6.3.3.1.3 Global topology generation

Global topology is generated by creating symbolic links between metric maps through switching nodes. For simplicity, all the edges of the topology map are assumed to have unit weights.

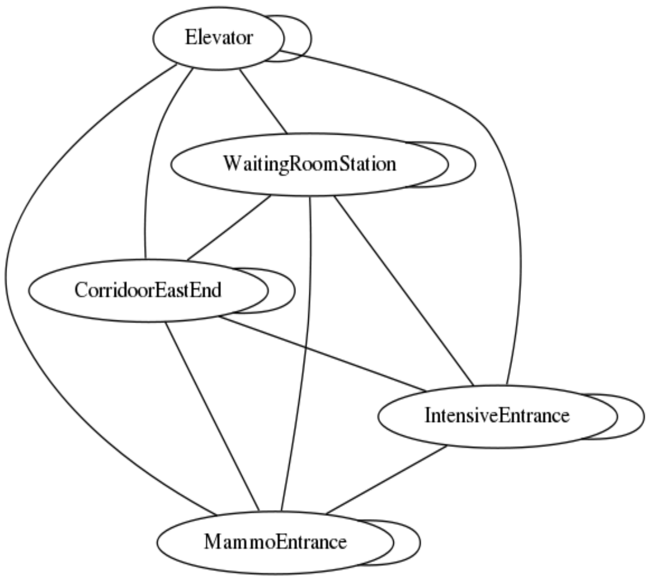

Figure 53, Local topology of the annotated map

Topology generation can be done online, because only the local topology information is used. The major advantage of online global topology generation is scalability. If a new annotated map of a new region is added to the existing set of annotated maps, global topology will be automatically extended and the robot will be able to reach to new points-of-interests in this region. Moreover, modifications to existing metric maps, such as adding or deleting places-of-interests, can be handled transparently, as the global topology is regenerated each time it is needed.

6.3.3.2 Navigation

Navigation process starts with a user request. The user specifies a destination for the robot from the list of all places-of-interest in the given set of annotated maps. Knowing the initial position of the robot on the global topology map, a global path is calculated using A* algorithm. The global path is a sequence of nodes, starting with the initial node, followed by a number of switching nodes (if the destination is not in the same local map), and destination node.

Initial position also determines the current local metric map. This is due to the fact that robot is initially at a particular place-of-interest, each place-of-interest is unique and only present in a single local metric map.

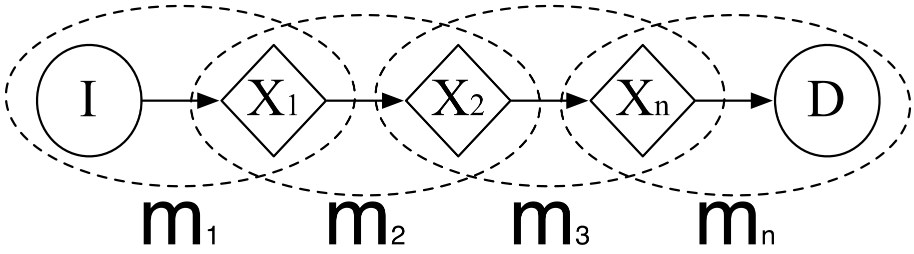

Figure 54, A global path consist of an initial node, a number of switching nodes and a destination node. Each two consecutive nodes reside in a single metric map

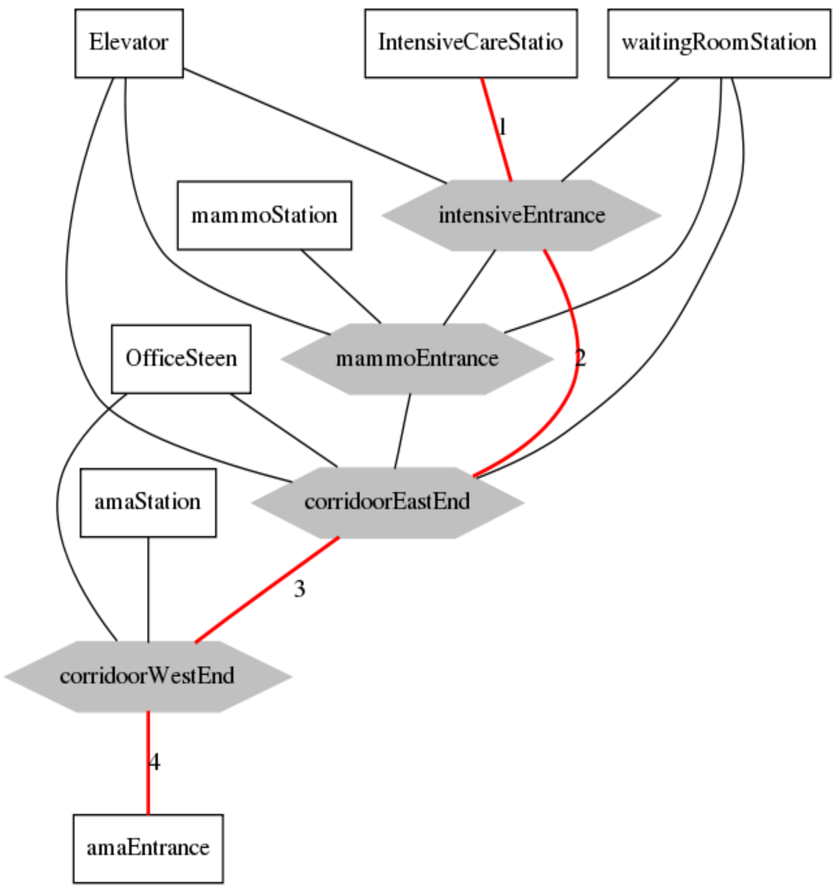

Figure 55, Global path from 'Intensive Care Station' to 'AMA Entrance'. Robot passes through 3 switching nodes; 4 local maps will be used

The second node in the global path is also in the same local metric map with the initial position. If the given destination is in the same local map with the initial position, global path is only composed of these two nodes. If the destination is not in the same local map with the initial node, the robot must pass through at least one switching node, as shown in Figure 54.

It is possible to plan and execute a local path from the first to the second node of the global path, using the local metric map. Robot can localize itself in the local metric map. As the robot approaches to the switching node, its expectation to observe the associated visual tag increases. Therefore, robot can determine whether a switching node is reached based on metric-map based localization and visual tag detection. When the robot reaches a switching node, the new local metric map is needed. Any two consecutive nodes in the global path can only exist in one single metric map, so the new local map can be traced by using the current and the next nodes in the global path.

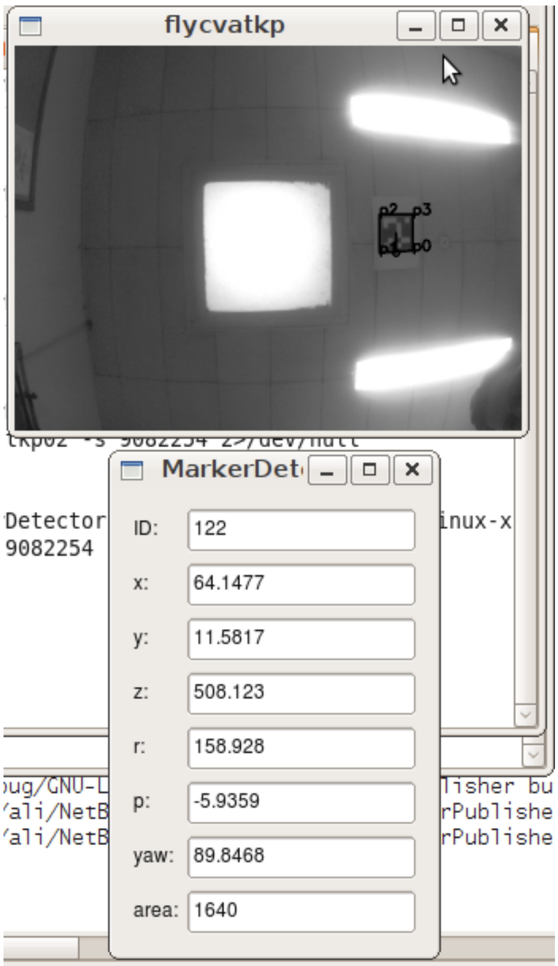

Figure 56, Artoolkit Plus [110] visual tags are detected at a switching node.

6.3.4 Experiments

The described method is developed as a software library and tested in a real hospital setting. CARMEN [183] is used as the software framework for the experiments. Gmapping [122] is used to generate metric maps. Based on a recent survey[184], ARToolKit Plus [110] markers (Figure 56) placed on the ceiling are used as visual tags at the switching nodes.

Physical experiments are conducted on a Pioneer-3dx platform, which is equipped with a Sick-LMS200 laser range finder, upward-looking PointGrey Firefly-MV cameras and a laptop computer running Linux operating system.

Experimental tests took place in the first floor of building 60, Bispebjerg Hospital, Copenhagen, Denmark. The environment is roughly 4500 m^2, and it is divided into five regions: Acute Medical Attention (AMA) unit, Offices, Waiting Room, Intensive Care Unit and Mammography Unit. Seven points-of-interest were identified in the environment, and there are four switching nodes in the global topologic map (Figure 55).

After initialization at a given place-of-interest, the robot was requested to randomly given places-of-interest in the environment. Several experiments were conducted, over a period of a month, at different times of the day, with different intensities of crowdedness.

Experiments demonstrated that the robot can successfully navigate in-between any point-of-interest inside the environment, handling the map transition effectively. It is also shown that providing an additional annotated map can easily extend the area covered for navigation. This map is handled automatically by the system, and becomes available immediately for navigation, due to online global topology generation.

A short video of the experiments can be seen in [185].

6.3.5 Discussion

The proposed scheme provides a practical framework for indoor mobile robot navigation, where the robot is required to operate between certain places of interest in the environment. This framework is targeted for service robot applications, and it is especially useful in logistics tasks such as [186]; where the robot is used for transportation of goods between different units of a hospital.

There are three salient advantages of the method. First, the method provides reliable navigation. The use of simple visual tags provide makes it possible to redundantly localize the robot. Visual tags are distinct and easily recognizable, allowing absolute localization in topology map. They also make it possible to improve metric localization, as they act as landmarks in the local metric map.

Secondly, the method allows scalable mapping and navigation. The use of the global topology map reduces memory requirements significantly because local metric maps are only symbolically linked through the topology map. New local maps can be easily added over the time, and the existing ones can be changed or modified individually.

Finally, the method can be easily adapted to new environments. Environmental modification is needed, but it is very easy and minimal. Building metric maps of smaller regions of the environment is also an easier task compared to mapping the entire environment, and visual tags will provide valuable information, especially in relatively featureless areas of the metric map.

6.3.6 Conclusion

In this paper, an indoor mobile robot navigation scheme based on the hybrid metric-topological maps is presented. The method is developed as a software library, and it was implemented on a physical robot for empirical evaluation in a real hospital environment.

The method is developed to address the navigational requirements of a service robot, and it provides a practical solution for indoor mobile robot navigation.

6.4 Automatically Annotated Mapping of Indoor Environments

Abstract—This paper presents a new method for mapping and annotating indoor environments. The method makes use of occupancy grid maps for metric representation, and Datamatrix barcodes to physically embed information in the environment. This method extends barcode detection to simultaneous localization and mapping in topological space, and fuses camera and robot poses to build an automatically annotated global topo-metric map. It is developed as a software framework and tested on a real robot in a typical indoor environment. Experiments show that the method is capable of producing globally consistent, automatically annotated hybrid metric-topological maps that can be used by a range of autonomous agents.

6.4.1 Introduction

Wide and ubiquitous availability of portable computational power has led to a great variety of navigational and location based services. There are already a number of widely used consumer-level applications, which provide navigation assistance or enable a range of services based on current location of the user. Many of these applications use GPS technology for localization, which works only outdoors. Even though GSM or WIFI triangulation techniques can be used in indoor environments, they can only provide a coarse estimate of position and orientation.

With wide applications, semantic mapping of man-made environments is evolving very fast, which meanwhile promotes many indoor location-based services. Indoor mapping has been a well-addressed problem, as an essential component for autonomous navigation of mobile robots. The state-of-the-art methods are capable of generating 2D metric representations of environments, which model the positions of objects and walls from a robot-centric perspective. However, for better perception from human-centric perspective, semantic information needs to be added. In most cases, the information is added manually. A handful of research and commercial projects have demonstrated the potential usage of indoor localization for location based services [187-190]. But because the indoor localization approaches in these applications require semantic information to be manually added to maps, which involve time-consuming preparation steps, they are considered to be not flexible enough to adapt to new working environments.

In this paper, in order to address the issue of manually adding semantic information, we propose a novel and integrated framework that provide metric, topological and semantic information about indoor environments. The framework integrates three major components: 1) simultaneous localization and mapping from robotics research, 2) object detection and camera pose estimation from computer vision, and 3) new generation barcodes from pervasive computing. These techniques span a wide range of applications and can be integrated to generate: 1) 2D metric maps of indoor environments, 2) physically annotated places, and 3) anchor relevant information using 2D barcodes. The resulting map is a robust and consistent representation of the environment that can be used in a wide range of applications; including e.g. service robotics, indoor navigation and location based services.

Annotation is one of the simplest forms of semantic information, which names certain places in the environment correspondingly. Especially in service robot applications, to facilitate human-robot interaction, annotation is a simple yet important technique to translate the robot-centric metric information to human-centric semantic information.

Our method relies on 2D barcodes that are installed in the environment. They are matrix-type barcodes that have relatively high data storage capacity and can encode information directly. They found a wide range of uses in industrial applications during the last decade. They are also extensively utilized in mass media and installed in public spaces for pervasive computing applications. (See Figure 1 for some examples of everyday applications that make use of 2D barcodes.)

In addition to 2D barcodes, our method uses occupancy grid maps to represent metric information of the environment, and topology maps to represent the connectivity of local information via points of interest, where each node of the topology graph corresponds to a barcode. The paper describes the method, which attempts to minimize the manual work that needs to be done for mapping, hide the complexity from the end-user for efficient utilization of the system, and provide a reliable and robust solution for navigation.

In general, buildings are designed according to architectural patterns to achieve optimum functionality, which means that certain places in the environment have more significance than others from the perspectives of the inhabitants (including robots in a broader sense). In this paper, we refer to these places as 'points-of-interest' (POI).

Figure 57, QR Codes link for schedules on bus stops in Osaka, Japan (top), Over 100,000 businesses in U.S. are identified as 'Favorite Places on Google', and each business receive a window decal with a 2D barcode that allows users to access reviews and business information (middle), Semapedia tag in Cologne airport, Germany (bottom). Semapedia project connects physical places to their relevant Wikipedia articles.

Determining which places in the environment are significant, or of interest for robots and humans is a work of the system designer. In the system design stage, inputs from users, administrators, robot designers and other stakeholders should all be considered to decide an optimal set of POIs. After this, spatial relationships between POIs, as well as their locations can be defined in a semi-automatic way, using the method proposed in this paper. The only interference of human is to lead the robot to explore the environment for once at the beginning. Information of POIs can then be acquired automatically, which will play a critical role to build both human-centric and machine-centric representations of environments.

The paper is organized as follows. Section II introduces related work in areas of visual tagging systems and robotic mapping. Section III explains our method in details. Section IV presents experimental verification of the method. Section V concludes the paper with discussions.

6.4.2 Related Work

6.4.2.1 Visual Tagging Systems

Wide use of barcodes started with their adoption into the retail industry. Universal Product Code [191] was the first standardized barcoding system that consisted of bars with varying widths in an alternating pattern. It is able to encode 13 digits of code including information about the physical origin of the product, the manufacturer and the product code.

However, UPC barcode was developed in a specific context and is not flexible for generic use. To address the shortcomings of the fixed lengths and numeric-only encoding, Code 39 [192] was developed. Code 39 (and its derivatives) is regarded as a machine-readable alphabet of a subset of characters from the standard ASCII table. 26 uppercase letters, 10 digits and 7 special characters can be encoded with Code 39, each represented as a set of 9 wide or narrow bars, 3 of which are always wide.

With the advancement of imaging systems 2D barcodes emerged in 90's. Their main feature is representing data with alternating contrasts of cells in a matrix form. Due to multi-dimensional representation, data capacity of 2D barcodes is significantly higher than 1D barcodes.

Two most widely used 2D barcode symbologies are QR Code [193] and Datamatrix [194], which were both designed for industrial purposes, such as tracking parts in industrial manufacturing [195]. Their public usage started with camera phones, which were used in mass printed media for marketing purposes

[196]. Today, 2D barcodes are being used in an extensive range of applications, some examples are shown in Figure 1.

The QR Codes and Datamatrix have very similar characteristics. They both can encode full ASCII set, and can be dynamically extended according to the size of the data. They also offer Reed-Solomon [133] error detection and correction capabilities (Figure 58).

A recent improvement on 2D barcodes is High Capacity Color Barcode (HCCB) [197], which use color as the third dimension to increase the data capacity without increasing the matrix size. On the other hand, detection performance of HCCB is more dependent on illumination, compared to conventional, singleton barcodes.

It is noteworthy to mention augmented reality applications, which have extensively employed visual tagging systems. Augmented reality is a term to define interactive applications, in which 3-D virtual objects are integrated into real environment imagery by using barcode-like markers as placeholders.

ARToolKit [144] project introduced template-based markers, together with an open-source framework for augmented reality applications. Markers consist of a representative pictogram and a thick-bordered square frame around the pictogram. Detection of markers is made possible by detecting the corners of the square frame, and searching for a previously learned pattern within the boundaries of the frame.

ID based markers ([110], [198]) are similar to template based ones, but instead of pictograms, they incorporate matrix-type patterns for identification. Dimensions of the grid define the maximum number of distinct markers, which is typically around a few thousand. In essence, they share the same characteristics with 2D barcodes, but their data capacity and encoding range is much lower.

Topology based markers ([147], [199]) are based on connected component analysis, and they are identified by the topology tree of its graphical elements and their connectivity. They significantly differ from template and ID based markers in the sense that there is no fixed frame structure and therefore two visually very different markers can actually correspond to the same topology tree.

In this work, we adopted Datamatrix [194] barcodes for physical linking of semantic information. It has very similar characteristics to QR Codes; such as extensibility and error checking and correction (as show in Figure 58). The main advantage of Datamatrix over QR Codes is the space efficiency [200]. Therefore, Datamatrix can be up to %60 smaller than a QR Code for the same data [201], [202].

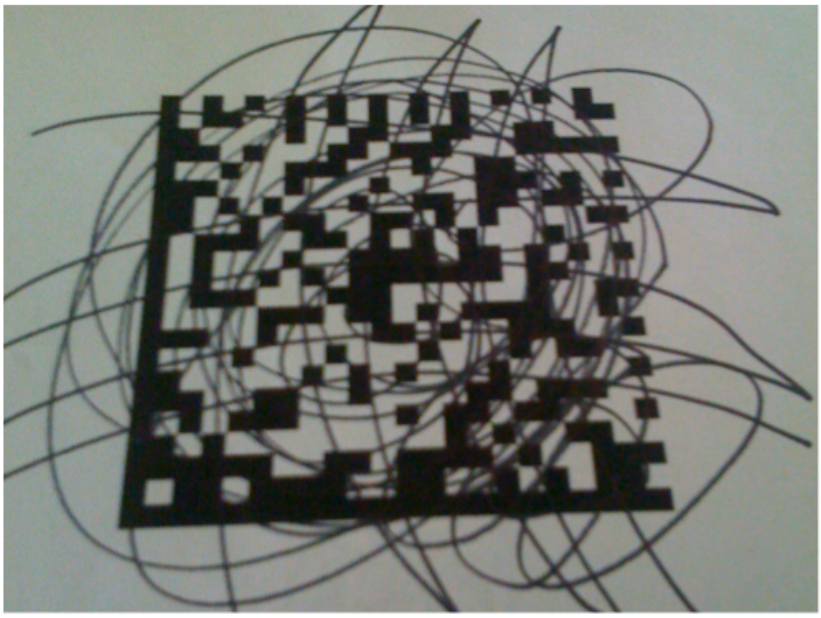

Figure 58, A Datamatrix [194]. This barcode can be still decoded, despite the partial corruption

6.4.2.2 Robotic Mapping

A number of map types are in use today for mobile robot navigation, which include metric maps [176], topological maps [177], sensor-level maps [178], appearance-based maps [179] and semantic maps [58].

Simply put, metric maps represent the environment in terms of distances that correspond to actual distances of the objects in the real environment. Occupancy grid maps [41] are the most commonly used metric maps, where the environment is represented as a matrix of cells with each cell possessing a value that corresponds to the probability of its being occupied.

In contrast, topology maps represent the environment in terms of its constituents and their connectivity without a necessity to use metric information. Therefore, topology maps mainly describe the topological structure of the environment, and how places or objects (nodes) are related to each other (edges).

Metric and topological maps are alternative ways of representing environments, and have complementary strengths and weaknesses. Metric maps are easy to build and maintain for small environments [45]. They can be used for accurate localization and optimal route planning. Metric maps also layout the environment in an easily readable way for humans [36]. Topological maps are very suitable for high level planning, and are easy to scale up for large environments. Sensor precision is not as important in the context of topological mapping, since there is no need for precise position estimation for map building [180]. It is also worthwhile to mention that topological maps are good representations for symbolic problem solvers.

Hybrid metric-topological maps combine these advantages by applying the appropriate types of map for different tasks. The seminal idea of hybrid maps was originally suggested in early works ([178], [58], [48]), with a number of applications (such as [46], [53], [52], [181], [182]) implemented during the last decade.

The need for including semantic information has been recognized for a long time [48]; especially when the robot needs to interact with the humans. While many robotic implementations incorporate some semantic information in their maps (e.g. [203], [204], [205]), this information is hand-coded into the system to provide additional information to navigation and localization modules.

In [206] and [207], two robot systems that can acquire semantic information through linguistic interaction with human are reported. These systems rely on speech synthesis and recognition to incorporate semantic information in their maps and communicate with human through semantic language.

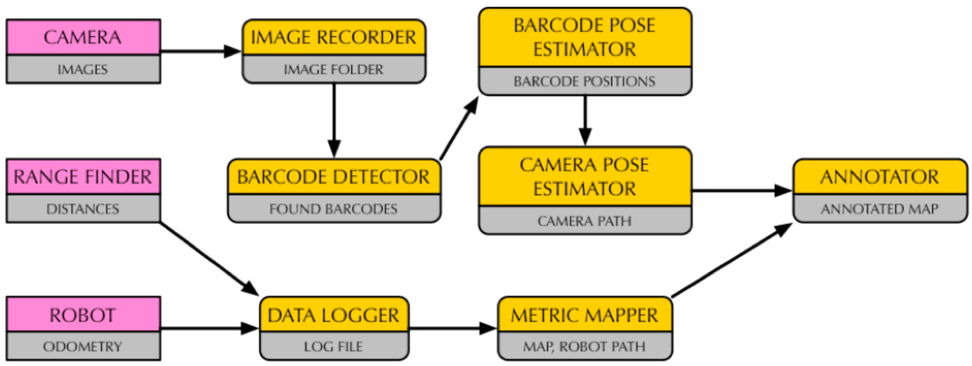

Figure 59, Flowchart of the method. Wheel encoders, laser range finder and camera are used to generate annotated maps

Most of the recent work on semantic mapping utilizes vision to classify the constituents of environments. In [205], spatial hierarchy of a domestic environment is anchored to the conceptual hierarchy based on visual identification of objects found in different rooms; e.g. a room is classified as a kitchen if a stove and a sink is detected. Yet, it is not addressed that how the knowledge of conceptual hierarchy is acquired by the robot (i.e. what objects should be expected in e.g. a kitchen). The method was tested in a home-like setting, and artificial objects (e.g. colored boxes and cylinders) were used to represent real house-hold items. A similar approach is presented in [208], where artificial markers in different rooms were used for place classification. Real objects were not used in experiments due to the fact that 'Reliable and robust methods for high-level feature extraction are not yet available'.

Our method differs from most of the recent semantic mapping approaches due to the fact that we choose not to employ place recognition. The reason for this choice is that we believe current technology is not mature enough to reliably solve the problem of ambiguity in real world application environments. In indoor environments such as hospitals, many places, such as doctor's offices, might fall into the same category if current techniques of place recognition are employed. As a result, extra information is needed to ensure a reliable navigation. In this work, we utilize direct annotation through 2D barcodes, and develop a method that can automatically generate the reliable semantic maps directly upon exploration. While high reliability of the method is assured, the extra manual work to deploy the system is also minimized, facilitating a commercialized application.

6.4.3 Method

The method consists of the following five steps:

- Visual tagging
- Exploration
- Topo-metric Mapping
- Relative Camera Pose Estimation
- Augmentation

Only steps 1 and 2 are carried out with involvement of the human (system) designer. The rest of the steps can be executed automatically. The flowchart given in Figure 59 illustrates the method from data acquisition to map generation.

6.4.3.1 Visual tagging

As an integral part of the actual system design, the points-of-interests (POIs) in the environment have to be determined beforehand. The task of determining POIs essentially belongs to the human designer of the overall system or application. This is also a heuristic process, usually based on knowledge about the architectural structure and the functional units residing in the environment. Once the POIs are determined, they are represented by corresponding Datamatrix barcodes and placed at their respective positions. We assume that every barcode has a single physical copy and it only exist at a unique POI in the environment. This is also called *physical linking* [209].

The information embedded in the 2D barcode depends on the application. In our method, apart from the location information, we encode the physical widths of the barcode on the printed media. This information can be used to estimate the relative pose of the camera (and therefore the robot) with respect to observed 2D barcodes. In succeeding sections on pose estimation, the detailed procedures will be elaborated.

Visual tagging is one of the two steps that involve human intervention, and may be the most time-consuming step in this approach. However, it is worthwhile to point out that time consumption of conducting this step is minimized due to the fact that the workers can install the visual tags in the environment with very coarse precision. The succeeding steps of pose estimation will automatically identify the précis pose of the visual tags, as well as the camera. This is also one of the major strengths of the method we proposed in the paper.

6.4.3.2 Exploration:

The main purpose of exploration step is to obtain the metric map of the environment. During exploration, the robot is manually guided in the environment to learn its characteristics. Three types of information are recorded during exploration: 1) readings from the wheel encoders of the robot base, 2) laser scans from a front-facing laser range finder, and 3) images from a monocular camera. All of the data are time-stamped for further processing.

In certain cases, the metric map of the environment might have already been available. In these cases exploration step becomes unnecessary, and the metric mapping step can be substituted for metric localization; as the rest of the method only needs the information of the path of the robot.

6.4.3.3 Topo-metric Map Generation

This step basically applies simultaneous localization and mapping (SLAM) and topological localization on the recorded dataset during exploration.

SLAM has been addressed by a large number of researchers, with several solutions gradually converged and came into dominant existence. In this work, we adopted the probabilistic framework of Rao-Blackwellized particle filters for occupancy grid mapping by Grisetti et.al. [63]. This framework uses raw odometry readings to estimate the motion of the robot and a history of laser range finder readings to refine the estimation. Our empirical study shows that it can provide robust solutions to our application environments.

Topological localization part of this step Correspond to scanning the image dataset to detect 2D barcodes that were observed during exploration. Each found barcode refer to a POI, and the information encoded in the barcode defines its characteristics. Processing the recorded image set reveals the topological path of the robot during exploration.. During *Augmentation* this information is needed to trace back the robot poses at the instants where a 2D barcode is visible.

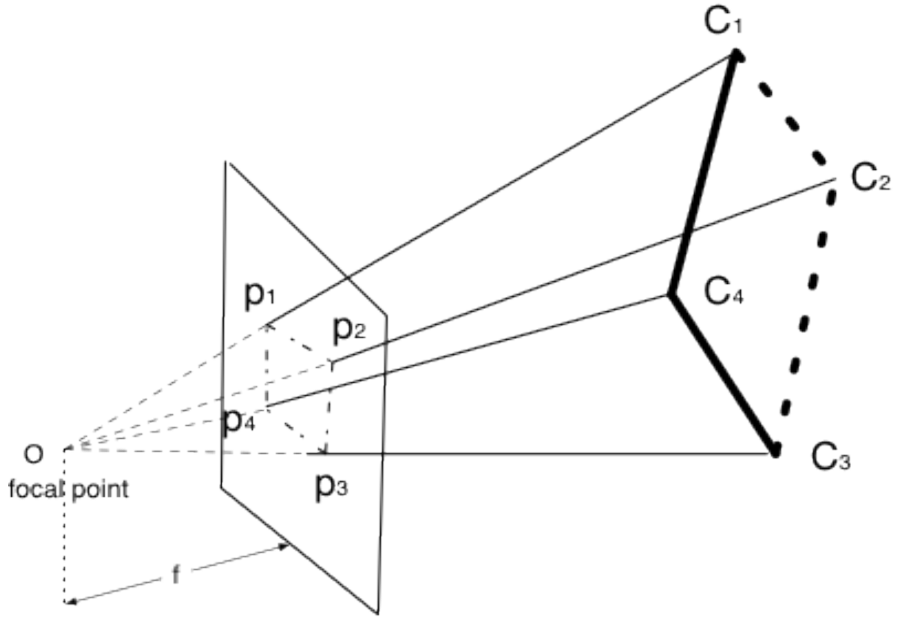

Figure 60, Pinhole camera model is used to project 3D points (barcode corners) onto image plane (2D points).

6.4.3.4 Camera Pose Estimation

Pose estimation is the problem of determining the position and orientation of a calibrated camera from a set of 2D-3D correspondences. In this method, we exploit two factors about the visual tags for pose estimation: the barcode shape, and the width information encoded on the barcode. It is noted that no precise position of the barcode is needed in the method, so we can actually put the barcodes in places with rough precisions. The 2D barcodes are designed around Datamatrix, a standard barcode symbology [194]. As seen in Figure 58, it has a square form with four distinct corners (left and bottom edges are solid whereas right and top edges has an alternating pattern).

Estimating the position and the orientation of the camera with respect to a scene from n 2D-3D correspondences is coined as *Perspective n-Point* (PnP) problem [210]. In our case, the problem reduces to 4 coplanar and non-collinear points, which yields to a unique solution. (For general solution of PnP, please refer to [211])

If we assume that 2D barcodes are placed on planar surfaces, four corners of the physical marker can be defined as:

Corner 0: (-w/2 , -w/2 , 0)

Corner 1: (-w/2 , w/2 , 0)

Corner 2: (w/2 , w/2 , 0)

Corner 3: (w/2 , -w/2 , 0)

where the origin of the global coordinate frame correspond to the center of the marker.

Figure 60 illustrates coplanar P4P problem. As the geometry of the barcode is known, the relation between physical dimensions of the barcode and real-world coordinates of its vertices can be written as the following:

$$\overrightarrow{C_1C_2} = \overrightarrow{C_4C_3} \Leftrightarrow \begin{bmatrix} X_{C_1} - X_{C_2} \\ Y_{C_1} - Y_{C_2} \\ Z_{C_1} - Z_{C_2} \end{bmatrix} = \begin{bmatrix} X_{C_4} - X_{C_3} \\ Y_{C_4} - Y_{C_3} \\ Z_{C_4} - Z_{C_3} \end{bmatrix} \tag{1}$$

Moreover, assuming pinhole camera model, we define the projection from real-world coordinates to image coordinates as:

$$X = Z\frac{1}{f_x}(u - c_x) \qquad Y = Z\frac{1}{f_y}(v - c_y) \tag{2}$$

Therefore (1) can be rewritten as:

$$\begin{bmatrix} Z_{C_1}\frac{1}{f_x}(u_{C_1} - c_x) - Z_{C_2}\frac{1}{f_x}(u_{C_2} - c_x) \\ Z_{C_1}\frac{1}{f_y}(v_{C_1} - c_y) - Z_{C_2}\frac{1}{f_y}(v_{C_2} - c_y) \\ Z_{C_1} - Z_{C_2} \end{bmatrix} = \begin{bmatrix} Z_{C_4}\frac{1}{f_x}(u_{C_4} - c_x) - Z_{C_3}\frac{1}{f_x}(u_{C_3} - c_x) \\ Z_{C_4}\frac{1}{f_y}(v_{C_4} - c_y) - Z_{C_3}\frac{1}{f_y}(v_{C_3} - c_y) \\ Z_{C_4} - Z_{C_3} \end{bmatrix} \tag{3}$$

Since $Z_{C_1} - Z_{C_2} = Z_{C_4} - Z_{C_3}$, we can simplify the above equation to:

$$\begin{bmatrix} u_{C_1}Z_{C_1} - u_{C_2}Z_{C_2} \\ v_{C_1}Z_{C_1} - v_{C_2}Z_{C_2} \\ Z_{C_1} - Z_{C_2} \end{bmatrix} \sim \begin{bmatrix} u_{C_4}Z_{C_4} - u_{C_3}Z_{C_3} \\ v_{C_4}Z_{C_4} - v_{C_3}Z_{C_3} \\ Z_{C_4} - Z_{C_3} \end{bmatrix} \qquad (4)$$

This equation correlates the depths of four corners of the barcode to its image coordinates. Assuming unit depth for Z_{C_1} simplifies equation (4) to:

$$\begin{bmatrix} u_{C_2} & -u_{C_3} & u_{C_4} \\ v_{C_2} & -v_{C_3} & v_{C_4} \\ -1 & 1 & 1 \end{bmatrix} . \begin{bmatrix} \dot{Z}_{C_2} \\ \dot{Z}_{C_3} \\ \dot{Z}_{C_4} \end{bmatrix} \sim \begin{bmatrix} u_{C_1} \\ v_{C_1} \\ -1 \end{bmatrix} \qquad (5)$$

where

$$\begin{aligned}
&\eta = (u_{C_3}v_{C_4} - v_{C_3}u_{C_4}) + (u_{C_4}v_{C_2} - u_{C_2}v_{C_4}) \\
&+(u_{C_2}v_{C_3} - u_{C_3}v_{C_2}) \\
&\dot{Z}_{C_2} = \frac{1}{\eta}(u_{C_1}(v_{C_3} - v_{C_4}) + v_{C_1}(u_{C_4} - u_{C_3}) \\
&-(u_{C_3}v_{C_4} - u_{C_4}v_{C_3})) \\
&\dot{Z}_{C_3} = \frac{1}{\eta}(u_{C_1}(v_{C_2} - v_{C_4}) + v_{C_1}(u_{C_4} - u_{C_2}) \\
&+(u_{C_4}v_{C_2} - u_{C_2}v_{C_4})) \\
&\dot{Z}_{C_4} = \frac{1}{\eta}(u_{C_1}(v_{C_2} - v_{C_3}) + v_{C_1}(u_{C_3} - u_{C_2}) \\
&-(u_{C_2}v_{C_3} - u_{C_3}v_{C_2}))
\end{aligned} \qquad (6)$$

Knowing the physical dimensions of the barcode:

$$\left|\overrightarrow{C_1C_3}\right| = \sqrt{(X_{C_1} - X_{C_3})^2 + (Y_{C_1} - Y_{C_3})^2 + (Z_{C_1} - Z_{C_3})^2} \qquad (7)$$

And correlations between depth and width (and height) from equation (2):

$$X \sim Z(u - c_x) \quad Y \sim Z(v - c_y) \tag{8}$$

We can obtain an expression that represents the depth of a corner of the barcode in terms of its physical width and image coordinates:

$$Z_{C_1} = \frac{\left|\overrightarrow{C_1C_3}\right|}{\sqrt{(\delta - 1)^2 + \phi^2 + \varphi^2}} \tag{9}$$

where

$$\begin{aligned} Z_{C_1} &= \delta . Z_{C_3} \\ \phi &= \delta(u_{C_3} - c_x) - (u_{C_1} - c_x) \\ \varphi &= \delta(v_{C_3} - c_y) - (v_{C_1} - c_y) \end{aligned} \tag{10}$$

Consequently, we can obtain X_{C_1} and Y_{C_1} by using equation (2), if intrinsic characteristics of the camera are known. Once the real-world coordinates of one of the corners is known, coordinates of the rest of the corners can be revealed by using equation (6).

Knowing the real-world position of the barcode makes it possible to estimate the pose of the camera with respect to the observed barcode. Assuming right-hand coordinate system at the center of the barcode, unit rotational vectors can be written as:

$$\vec{r_\alpha} = \frac{\overrightarrow{C_1C_2}}{\left|\overrightarrow{C_1C_2}\right|} \quad \vec{r_\beta} = \frac{\overrightarrow{C_1C_3}}{\left|\overrightarrow{C_1C_3}\right|} \quad \vec{r_\gamma} = \vec{r_\alpha} \wedge \vec{r_\beta} \tag{11}$$

where and $\vec{r_3}$ is the orthogonal unit vector to $\vec{r_1}$ and $\vec{r_2}$.

Finally, the distance between the camera and the barcode gives the translation of the camera, where the center of the barcode (C_O), which is the median point of C_1, C_2, C_3, C_4:

$$\vec{t} = \begin{bmatrix} (X_{C_1} + X_{C_2} + X_{C_3} + X_{C_4})/4 \\ (Y_{C_1} + Y_{C_2} + Y_{C_3} + Y_{C_4})/4 \\ (Z_{C_1} + Z_{C_2} + Z_{C_3} + Z_{C_4})/4 \end{bmatrix} = \begin{bmatrix} X_{C_O} \\ Y_{C_O} \\ Z_{C_O} \end{bmatrix} \quad (12)$$

6.4.3.5 Global Pose Estimation

The global pose of a visual tag can be deduced through the combination of the robot pose (3D) and the camera pose/orientation (6D) with respect to the barcode (As illustrated in Figure 61). The following is a more detailed description of the method.

Given the robot pose at time t_r

$$p_{robot,t_r} = \begin{bmatrix} x_{robot,t_r} & y_{robot,t_r} & \theta_{robot,t_r} \end{bmatrix}^T \quad (13)$$

And the camera pose at time t_c

$$p_{cam,t_c} = \begin{bmatrix} x_{cam,t_c} & y_{cam,t_c} & z_{cam,t_c} & \alpha_{cam,t_c} & \beta_{cam,t_c} & \gamma_{cam,t_c} \end{bmatrix}^T \quad (14)$$

The global marker pose can then be estimated as

$$p_{barcode} = \begin{bmatrix} x_{robot,t_r} + x_{cam,t_c}\cos(\alpha_{cam,t_c}) \\ y_{robot,t_r} + y_{cam,t_c}\sin(\beta_{cam,t_c}) \\ z_{cam,t_c} \\ \theta_{robot,t_r} + \alpha_{cam,t_c} \\ \beta_{cam,t_c} \\ \gamma_{cam,t_c} \end{bmatrix} \tag{15}$$

If we assume that $\Delta t = |t_r - t_c| < \xi$ is a time constant that is sufficiently small. In other words, the motion between recording an odometry message and recording an image is neglected, if the time difference between messages is sufficiently small. In our experiments, we found that $\Delta t < 0.2s$ gives satisfactory results for a robot with maximum speed of 1 m/s.

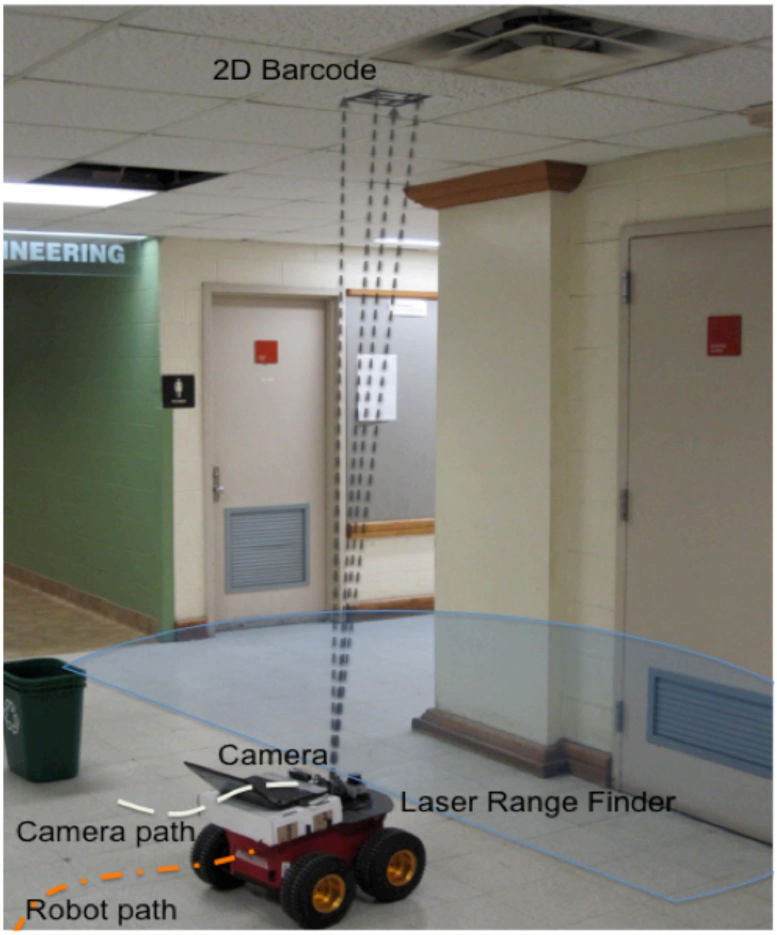

Figure 61, Two pose estimations are fused to generate the hybrid map

6.4.4 Evaluation

6.4.4.1 Implementation

Several open source libraries are used in implementation of the method. For 2D barcode generation, libdtmx library is utilized [212]. After determining POIs in the environment, we placed the printed out markers to their respective positions manually. It is worthwhile to point out that barcodes are attached to the ceiling, as the experiments were held in a real hospital and ceiling provided the best visibility of the barcodes in the least intrusive way.

The CARMEN [107] framework provides a software interface to record measurement data and link sensors to the robot base. During exploration, the robot was manually guided through the environment. Along with odometry and raw laser scan data; images are also timestamped and recorded. Consequently, metric maps are generated using Gmapping [122]. Together with the metric map, the path of the robot is obtained; which is later used to augment the relative camera pose estimations onto global metric map.

For topology generation, libdmtx [212] is used to detect the barcodes in the recorded image dataset. After scanning the dataset, a number camera poses are estimated, and an average pose is calculated for each barcode.

Finally, the corrected robot path and camera pose estimates are fused to obtain global pose estimates of 2D barcodes, by matching path and pose data that have the closest timestamps, as described in Section III.F.

At the end of the process, following are obtained:

- Metric map of the environment
- Connectivity graph of the visited nodes
- Annotated topo-metric map of 2D barcodes

6.4.4.2 Environment

Experiments took place on the 5th floor of Steinman Hall of the City College of New York. Nine locations were determined as the points of interest. Barcodes were printed on plain US-letter sized paper and attached to their respective locations.

6.4.4.3 Dataset

The robot traveled 127.9 m in 625 seconds. During the experiment, 845 images (8-bit, single channel, VGA resolution) and 5741 pairs of odometry and laser scan messages are recorded.

6.4.4.4 Topo-metric Mapping

Metric mapping was carried out off-line on a laptop with 2.53 GHz Core 2 Duo Processor and 4Gb RAM. Altogether 15 particles were used to generate the map. The filter only incorporated laser scans if the robot had moved more than 0.4 meters or 0.5 radians between consequent poses. It took approximately 134 seconds to generate the map (with a resolution of 0.1m/pixel) as well as the robot path from the raw dataset.

All of the images in the dataset were scanned for visual tags. An average scan took 0.874 seconds (716 seconds for the whole dataset). If a barcode was found in an image, camera pose estimation had to be carried out consequently.

In total, 70 barcode were detected in the dataset, covering all of the placed markers. Each marker was detected 8 times in average (min. 3, max. 18 times).

6.4.4.5 Global Pose Estimation

Each camera pose estimate inherits the timestamp of the image that includes the 2D barcode. For each camera pose estimate, the robot path is traced to find the robot pose data that has the closest timestamp (that is smaller than Δt =0.2 seconds), and they are regarded as a match. This leads to a number of pose estimates, which are then averaged to a single global pose estimate for each barcode.

6.4.4.6 Results

We used the architectural floor plan of the 5th floor as the reference and aligned it over the generated map of the environment (Figure 6). We measured the maximum deviation of the robot-generated map from the reference as 12 pixels (which correspond to 1.2 meters).

Maximum deviation in position is calculated as 0.4433 meters at *'Depot'* (out of 10 occurrences), and maximum deviation in rotation is 0.04422 radians at the Robotics Lab (out of 6 occurrences).

6.4.5 Discussion

The results show that, the method for building topo-metric maps is capable of generating consistent and accurate maps of the environment that are automatically annotated.

Metric portion of the map is compared with the floor plan, and it is found that the maximum error is 1.2 meters (from a real wall to an estimated wall) over a path of 127.9 meters and in an area of approximately 40x40 meters.

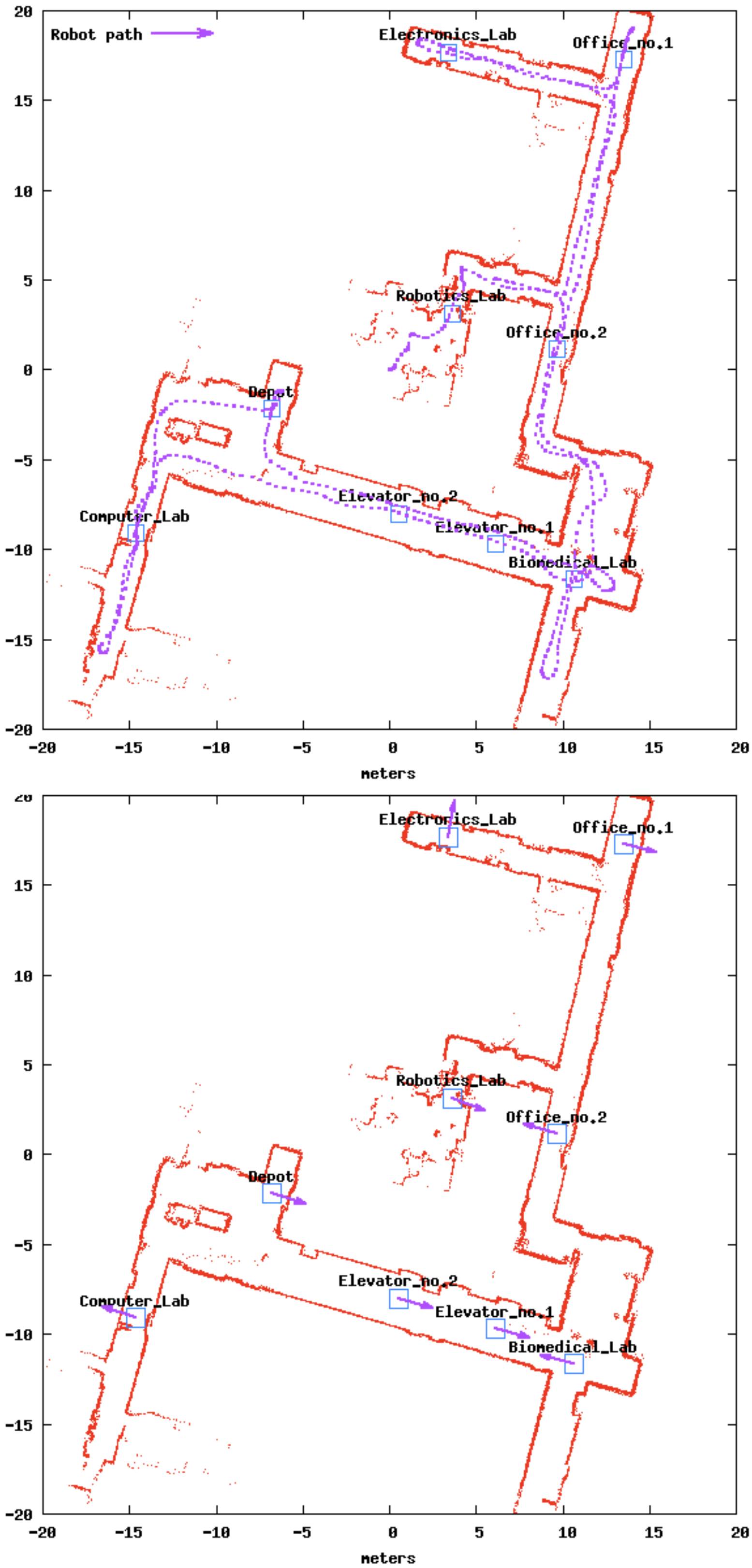

Figure 62 Top: Path of the robot during exploration. Bottom: Resulting annotated map. Each POI is captured during exploration, augmented on the metric map with respect to their global pose.

Figure 63, Generated map is overlayed on the architectural floor plan of the environment for comparison.

Physical characteristics of the camera and the barcode; such as camera resolution and calibration, physical size of the barcode and distance between the camera and the barcode has an effect on the accuracy of the results. Errors resulting from physical characteristics can be improved by e.g. using a larger barcode or a higher resolution camera.

Synchronization differences between robot pose estimates and camera pose estimates can also result in inaccuracies. It is not always possible to record odometry data and images synchronously, and robot might have slightly moved during the recording of an image. Due to this fact, that we imposed a maximum timestamp difference measure ($\Delta t = 0.2$ *seconds*). In other words, an odometry message and an image (in which a barcode is detected) are considered as a match if the difference between their timestamps is less than Δt. The maximum velocity of the robot during exploration is set as *1m/s,* therefore maximum estimation error due to synchronization difference is going to be 0.2 meters. Lower velocities will increase the exploration time, whereas smaller Δt will improve accuracy in the expense of fewer matches between robot poses and camera poses.

The effect of this phenomenon can be seen in Figure 64. As the robot moved along the corridor near the POI *Computer_Lab* several camera pose estimates were recorded. Because the robot passed by the 2D barcode in two opposite directions

(North-South, South-North) during exploration, the resulting camera pose estimates form two bundles. Interestingly, mean global pose estimate is accurate because synchronization errors cancel-out each other as the camera sees the barcode from two different directions. In general, it is found that global pose estimates tend to be accurate, but not necessarily very precise if the target is seen from multiple orientations.

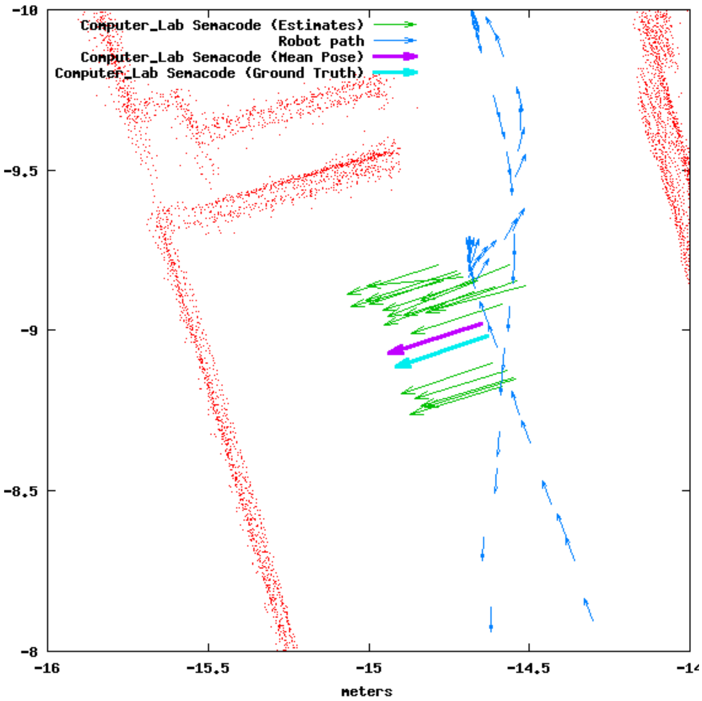

Figure 64, The metric map, robot path and barcode estimates around *Computer_Lab*. The barcode is perceived from two different directions during exploration.

6.4.6 Conclusion

In this paper, we introduced a new topo-metric mapping method for indoor environments. The method is principally developed for indoor service robotics applications, and aims to minimize the manual work that needs to be done by the human designer of the system.

The main contribution of the method is automated annotation of points of interest via 2D barcodes. The resulting map can be useful for both robot-centric tasks (localization, motion planning) and human-centric tasks (symbolic problem solving, assistive navigation for blind or visually impaired people) that require semantic information.

The method utilizes 2D barcodes to physically annotate places; metric SLAM to generate metric maps; and camera pose estimation to augment annotations to the global metric frame. It is implemented on a real robot and evaluated with physical experiments. The results show that the method can produce consistent and accurate annotated maps of the environment.

6.5 Mapping of Multi-Floor Buildings: A Barometric Approach

Abstract— This paper presents a new method for mapping multi-floor buildings. The method combines laser range sensor for metric mapping and barometric pressure sensor for detecting floor transitions and map segmentation. We exploit the fact that the barometric pressure is a function of the elevation, and it varies between different floors. The method is tested with a real robot in a typical indoor environment, and the results show that physically consistent multi-floor representations are achievable.

6.5.1 Introduction

The problem of simultaneous localization and mapping is one of the most researched topics in robotics. Many methods are proposed to solve the SLAM problem, and it is generally accepted that the problem of mapping indoor environments is well addressed. However, a vast majority of the state-of-the-art methods do not incorporate global constraints, and they only address the problem in 2D. Therefore, mapping of individual floors of a building is considered to be solved, but providing a globally consistent and well-aligned map of the whole building remains as an open question.

In this paper, we aim to tackle this problem using a sensor fusion approach. We exploit the properties of barometric pressure to detect floor transitions of the robot during exploration, and to situate individual floors of the building onto a global coordinate frame.

The major contribution of this paper is a simple, inexpensive and easy-to-implement approach for generating globally consistent maps of multi-floor buildings. Resulting representations can be called as 2.1-dimensional; as the individual floor maps are 2D occupancy-grids and floor elevations are represented as barometric pressures, which are proportional to actual elevation differences between floors.

Our method is built on top of publicly available, and open-source hardware and software frameworks, so that it can be easily implemented on existing systems. We implemented our method into a mobile robot system and evaluated the method in a typical multi-floor building. Results of the experiment prove that the method is capable of generating consistent maps for multi-floor buildings.

6.5.2 Related Work

Mapping of indoor environments is a well-studied problem; several solutions exist for 2D maps and their extension to 3D are emerging. According to Busckha[36], robotic mapping methods fall into 5 categories: metric maps [176], topological maps [177], sensor-level maps [178], appearance-based maps [179] and semantic maps [58]. In this paper, we focus on (metric) occupancy-grid maps, which is the most commonly used type of maps in robotics.

Majority of occupancy-grid mapping techniques utilize probabilistic approaches to model robot motion and to represent sensor uncertainties, such as derivations of Kalman filters ([213], [72]), particle filters [214] and graph-based optimization methods [215].

Despite the number of solutions for 2D mapping, only a few methods deal with mapping of multi-floor buildings. In [216], a visual-odometry based approach is proposed to derive the alignments of floors. In this approach, however, the floors are needed to be mapped sequentially; and salient visual features are required during floor transitions.

SLAM using multiple robots has similar aspects to the problem of mapping multi-floor buildings. In [217], a map-merging approach is presented to obtain a global map from the individual maps of robots, provided that robots meet with each other during their mission. In [218] this approach is extended in a way that the robots localize each other in their respective maps and create constraints that are used to merge the maps via global localization.

A recently proposed a method by Karg et.al. [219] is one of the very few methods that directly address the multi-floor mapping problem. Similar to [218], they use constraints that are derived from global localization to align maps of individual floors. Their method is suitable for multi-robot teams, and it is based on the assumption that certain architectural features of the building are common between different floors. A single sensor modality (laser range finder) is used for global localization and generation of constraints. This method does not perform well if certain architectural features are repetitive, symmetrical or dis-similar in different floors. Moreover, it does not address the issue of which map correspond to which floor of the building, and therefore requires this information has to be manually given.

Our method differs significantly from the above-mentioned methods, both in the way the maps are generated and in the way they are represented. Instead of only providing a correct alignment of the floors, his method also generates a globally consistent representation of the building; which directly correspond to elevations of individual floors, and therefore the actual layout. Furthermore, it does not depend on architectural similarities between different floors, as 2D SLAM is actually sufficient for alignment. On the other hand, our method currently supports single robot systems due to the fact that maps of individual floors are segmented from a continuous, global dataset.

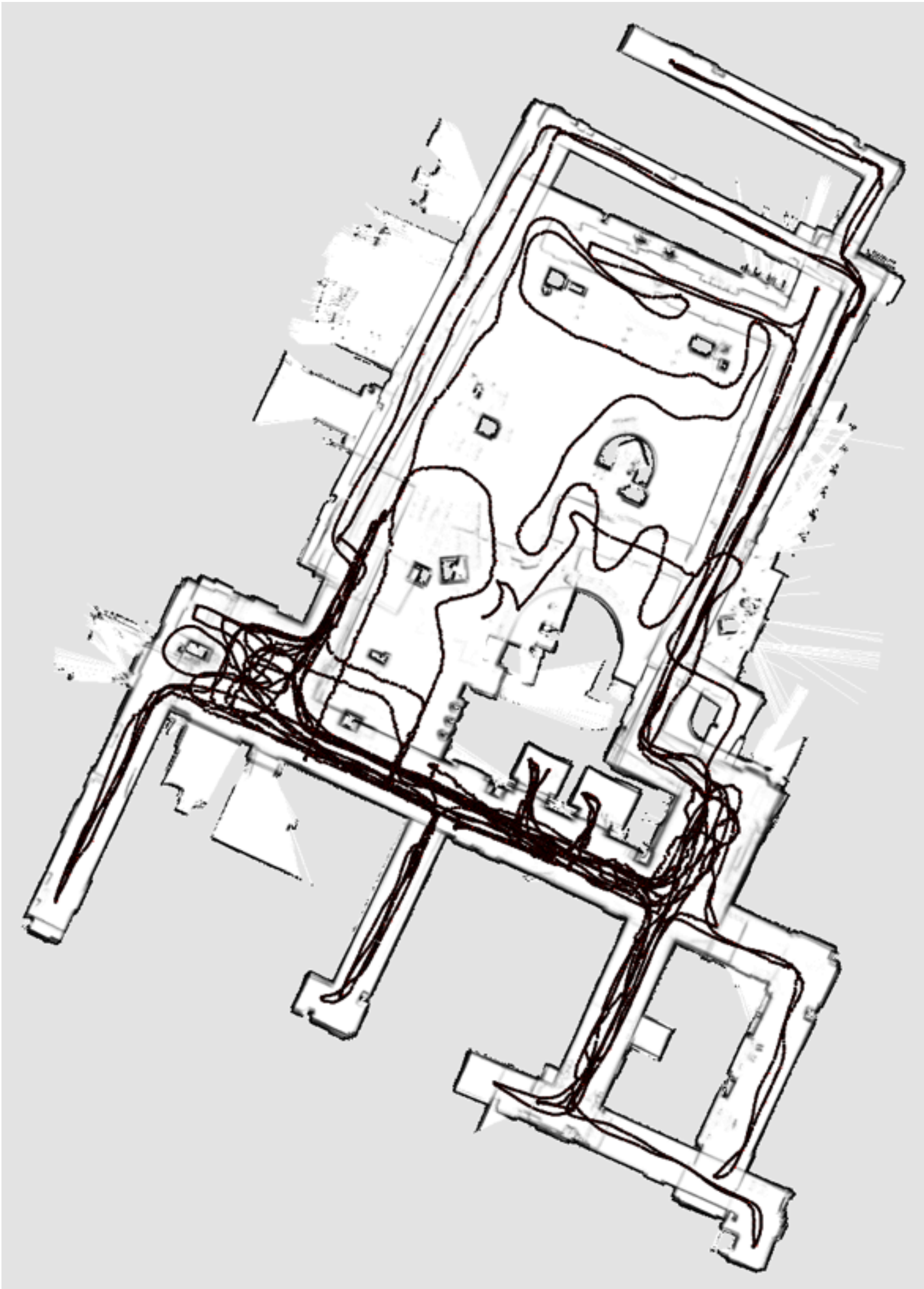

Figure 65, Map of a multi-floor building, as the result of 2D-SLAM: Six floors of the same building are superimposed onto a single plane. The dataset is recorded by a single robot exploring the whole building. The main goal of this work is to segment the map using variations of barometric pressure among different floors.

6.5.3 Method

We combine a laser range sensor and a barometric pressure sensor to align individual maps, detect transitions, and represent floor elevations in a global coordinate frame. Using a laser range finder, we generate 2D-maps of floors and estimate robot path during exploration, and floor transitions are detected using a barometric pressure sensor.

6.5.3.1 Barometric Pressure vs Altitude

Barometric pressure (or atmospheric pressure) is defined as the force per unit area exerted against by the weight of air above that surface in the earth's atmosphere [220]. Therefore, the change in pressure over an infinitesmall change in altitude should be proportional to the gravitational force exerted by the mass of the air in that infinitesmall layer. This relation can be expressed as:

$$\frac{dP}{dz} = -\rho g \qquad (1)$$

where P is pressure, z is altitude, ρ is density of air and g is the gravitational constant. The negative sign denotes the decrease in pressure with increasing altitude.

Furthermore, ideal gas law states

$$P = \rho R T \qquad (2)$$

where R is the *Boltzmann* constant and T is the temperature.

Therefore, using (2) and (3), we can write

$$\frac{dP}{dz} = -\frac{g}{RT} P \qquad (3)$$

According to the International Standard Atmosphere Model formulated by International Civil Aviation organization [221], zero altitude is measured from mean sea level as

$$P_0 = 101325 \, Pa \qquad (4)$$

Therefore, for constant gravitational acceleration and temperature, it is proven that the altitude can be approximated in terms of pressure, using the first order integral of equation (3):

$$z = -\frac{RT}{g}\log\left(\frac{P}{P_0}\right) \quad (5)$$

Our method basically exploits the fact that the barometric pressure decreases as the robot moves to upper floors, and increases as it moves to lower floors. Furthermore, we assume that the pressure remains relatively stable and constant in the same floor, and the pressure difference between the floors is significant between adjacent floors.

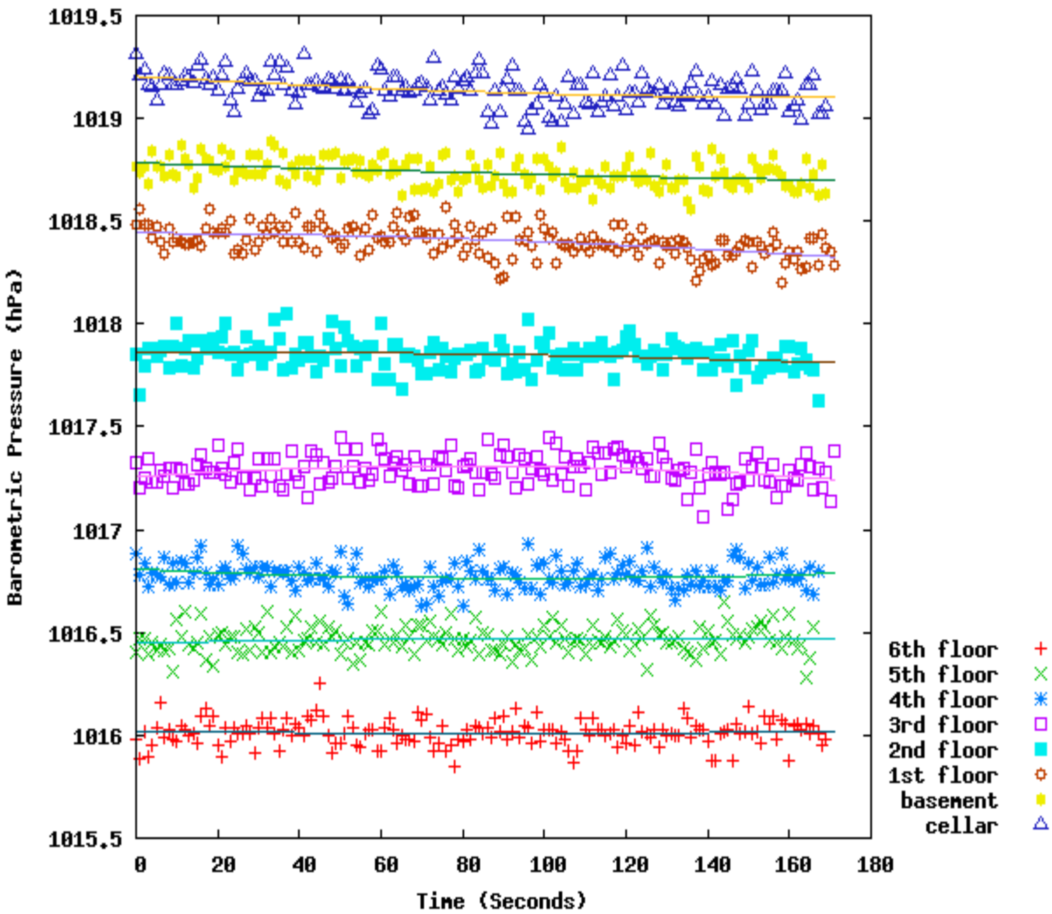

Figure 66, Absolute pressure readings from 8 different floors of a building.

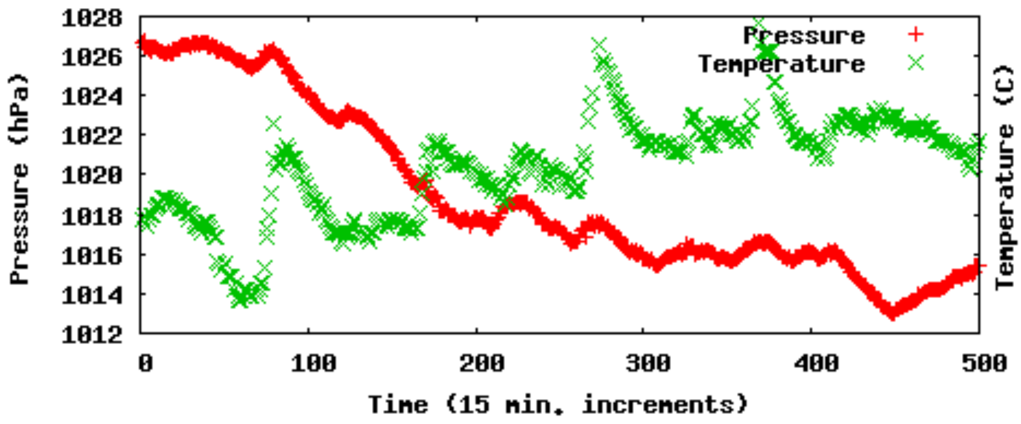

Figure 67, Change of the barometric pressure and the temperature over 5 days at a static point. Atmospheric events affects the density of air, hence the barometric pressure.

To verify these assumptions, the following experiment was conducted. Using a digital barometric pressure sensor, we recorded pressure and temperature from 8 different floors of a typical building. In each floor, data was recorded for the same amount of time (3 minutes), with refresh rate of 1 measurement per second. (Figure 66)

Two observations can be made from this experiment; during short periods (as in floor transitions), pressure remains relatively stable in the same floor, and the pressure difference is significant between floors.

To further prove that the barometric pressure at a static observation point remains relatively same over short periods of time, we collected temperature and pressure readings from a static point for 5 days (Figure 3). Barometric pressure varies due to atmospheric events, day-night transitions or geographical conditions. Yet, the variations in the barometric pressure happen relatively slowly, as seen in Figure 67. Considering the fact that moving from one floor to another (using elevators) takes at most a few minutes, it can be assumed that the barometric pressure at the respective floors remain the same over the period of transition.

Using these assumptions, the following sections deal with the detection of floor transitions based on barometric pressure changes between floors.

6.5.3.2 Exploration

Our method is developed for a single robot that explores the multi-floor building. During exploration, three types of data are recorded:

- Raw odometry from wheel encoders
- Range readings from laser range finder
- Pressure readings from barometric pressure sensor

At the end of the exploration, we obtain a single dataset that contains time-stamped readings of above-mentioned data.

6.5.3.3 *Mapping of individual floors*

Based on the state-of-the-art algorithms that address 2D mapping problem, we assume that sufficient solutions exist to generate metric gridmaps of individual floors. Particularly, we adopted the approach of Grisetti et.al. [64] that is based on Rao-Blackwellized particle filters[62], and made available in [122] as an open-source library.

To mention briefly, the key idea of Rao-Blackwellized particle filter for SLAM is to estimate the joint posterior $p(x_{1:t}, m \mid z_{1:t}, u_{1:t-1})$ for the map *m* and the path $x_{1:t} = x_1, \ldots, x_t$ of the robot [62]. Using the observations (range readings) $z_{1:t} = z_1, \ldots, z_t$ and the odometry measurements $u_{1:t-1} = u_1, \ldots, u_{t-1}$, it is possible to estimate the joint posterior, by using the following factorization:

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t-1}) = p(m \mid x_{1:t}, z_{1:t}).p(x_{1:t} \mid z_{1:t}, u_{1:t-1}) \qquad (6)$$

Using equation (6), it is possible to first estimate the path of the robot $p(x_{1:t} \mid z_{1:t}, u_{1:t-1})$, and consequently build the map of the environment $p(m \mid x_{1:t}, z_{1:t})$, based on the estimated path.

We use Gmapping [122] to process the dataset and to obtain the corrected robot path based on laser readings. At this point, the whole dataset is processed and it is treated as it is from a single floor rather than a multi-floor environment.

The resulting map is based on the path of the robot, and it is essentially the projection of all floors onto a single plane (As seen in Figure 65).

6.5.3.4 *Map segmentation*

Map segmentation is the process of dividing the monolithic dataset into segments to represent individual floors. Our method only takes barometric pressure readings into account for map segmentation; and unlike the available methods, repetitive, symmetrical or dissimilar architectural features in the environment do not affect the performance.

Segmentation is based on detecting discontinuities of the pressure graph (e.g Figure 68). Raw pressure readings are noisy time series, and for better representation, the raw data is smoothed using Double Exponential Smoothing [222]. The following two equations are associated with Double Exponential Smooting:

$$S_t = \alpha . y_t + (1-\alpha).(S_{t-1} + b_{t-1}) \qquad 0 \le \alpha \le 1$$
$$b_t = \gamma .(S_t - S_{t-1}) + (1-\gamma).b_{t-1} \qquad 0 \le \gamma \le 1 \tag{7}$$

where α and γ are smoothing constants, y is the raw data, b is the trend of the data and S is the smoothed data. The initial values for the smoothed data and the trend can be taken as $S_1 = y_1$ and $b_1 = y_2 - y_1$ [223].

The detection of floor transitions is based on the rate of change of the smoothed pressure readings. Figure 68 (bottom) shows the rate of change in pressure, i.e. $\Delta P = |P_t - P_{t-1}|$. Using the mean ($\mu$) and standard deviation (σ), floor transitions are detected at the instants where $\Delta P > \mu + 2\sigma$.

Detecting the segments based on the barometric pressure reveals when the robot travels from one floor to another. Using the timestamps of transitions, the whole dataset is divided into segments of odometry, range and pressure readings from individual floors. We apply SLAM on the segments to generate 2D maps, and we calculate the mean pressure for each floor.

The resulting representation is a globally consistent multi-floor map of the building. It can be referred as a 2.1D map, since the individual floors are represented as 2D metric maps, and the elevations of the floors are represented in terms of pressures.

6.5.4 Evaluation

6.5.4.1 Environment

We tested the method in a typical multi-floor building using a mobile robot. The experiment took place in Steinman Hall of the City College of New York. The building houses several engineering departments, and it is mainly consisted of offices, classrooms and laboratories.

During the experiment, the robot explored publicly accessible areas of the building. Office layouts and laboratory arrangements differ significantly between departments and therefore between individual floors.

6.5.4.2 Robot platform

The robot used in the experiment consisted of a Pioneer P3AT robot base, a SICK-LMS 200 laser range finder, an SMC1000 barometric pressure sensor and a laptop with Linux operating system. The pressure sensor was interfaced to the laptop through an Arduino microcontroller board, via USB.

6.5.4.3 *Exploration*

Experiment started at the 5th floor, where the CCNY Robotics Lab is situated. After covering the 5th floor, the robot traveled to 6th,4th,3rd,2nd and 1st floors of the building.

The total length of the path the robot traveled is approximately 927 meters. The overall exploration, including transitions between floors using elevators took 46.4 minutes, resulting in an average speed of 0.33 meters per second.

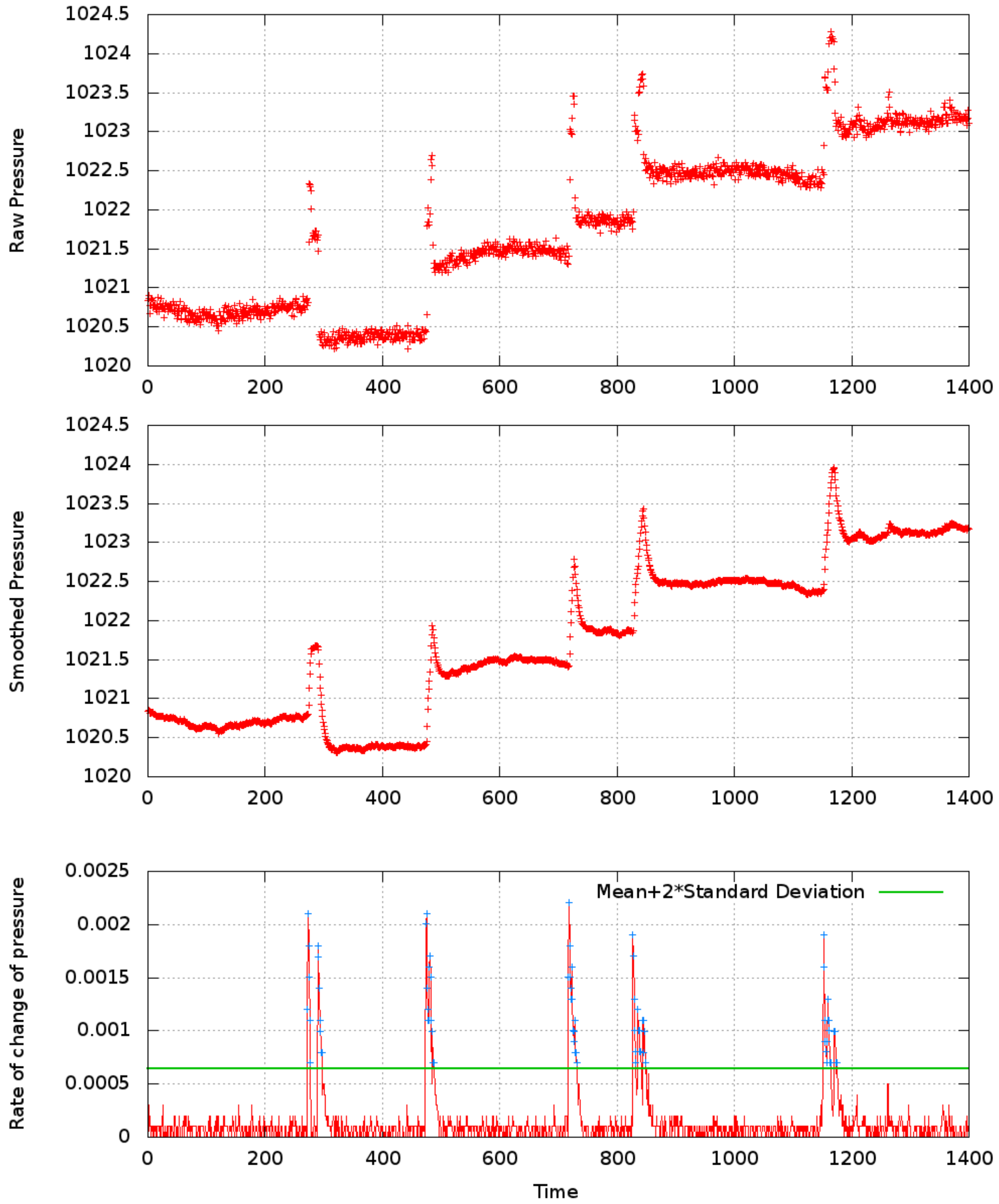

Figure 68, Barometric pressure during exploration. The robot starts its route at the 5th floor, then moves to 6th, 4th, 3rd, 2nd and ground floors. Raw pressure readings (top) are smoothed using Dual Exponential Smoothing (middle). The rate of change of the smoothed pressure reveals the instances for floor transition (bottom).

Figure 69, Steinman Hall - Grove School of Engineering of the City College of New York, City University of New York. Six floors of the building – Ground floor to 6th floor- are mapped.

6.5.4.4 *Metric Mapping*

We applied Rao-Blackwellized particle filter to the whole dataset using [122]. As a result of the SLAM, the path of the robot and the map are estimated (Figure 65).

Resulting map is the projection of all floors of the building on a plane: A superimposed representation of a 3D environment into a 2D map.

The only commonly covered parts of the individual floors are the main aisles, where the elevators are located. When the robot moves from one floor to another; it observes the main aisles before and after entering elevators.

6.5.4.5 *Map Segmentation*

In order to segment the map obtained in the previous step, we use barometric pressure data that is recorded during exploration. Figure 68 illustrates the change of barometric pressure among different floors. The exploration starts at the 5th floor, where the Robotics Lab is situated. After covering the 5th floor, robot takes the elevator to the 6th floor, where the pressure is lower as expected. Similarly, the barometric pressure increases as the robot moves to lower floors.

Transition of the robot between floors is clearly observable from the barometric pressure data. The elevator is a closed chamber, where a heating, ventilating and air conditioning (HVAC) unit controls the air circulation inside the cabin. The effect of this phenomenon can be identified as spikes on the graph.

The result of map segmentation process can be seen in Figure 70. Based on the discontinuities of the barometric pressure data; the dataset is segmented into six consecutive parts, and maps of six individual floors are obtained.

As iluustrated, the ground floor is a large open space, and the rest of the floors are arranged differently, based on the functional units residing in that floor. The common area in all of the floors is the main aisle, where the elevators are located.

To obtain a physically consistent representation, we directly use the mean barometric pressure for each floor as a measure of elevation. Therefore, 2D layout of the environment is represented as a metric X-Y grid and computed by the SLAM; and the elevation (Z) of the each grid represented by mean barometric pressures that are directly derived from the dataset.

6.5.5 Discussion of Results

The results of the experiment are illustrated in Figure 70 and Figure 71; which shows that the method is useful for building physically consistent maps of multi-floor environments.

The method is based on a set of very simple rules. It uses state-of-the-art metric mapping techniques, and exploits the characteristics of barometric pressure. In contrast to maps generated in [219], where the floor levels are hand-coded into the system; this method is capable of representing the correct order of the floors automatically. Moreover, the elevation differences between floors can also be revealed. From Figure 71 (middle), it is possible to observe that the floors are not evenly spaced. Larger elevation differences between the ground floor, the 2nd floor and the 3rd floor are consistent with the architecture of the building: The ceiling on the ground (1st) floor is significantly higher, and there is a mezzanine floor (2M) between the 2nd and the 3rd floors, which is not explored by the robot.

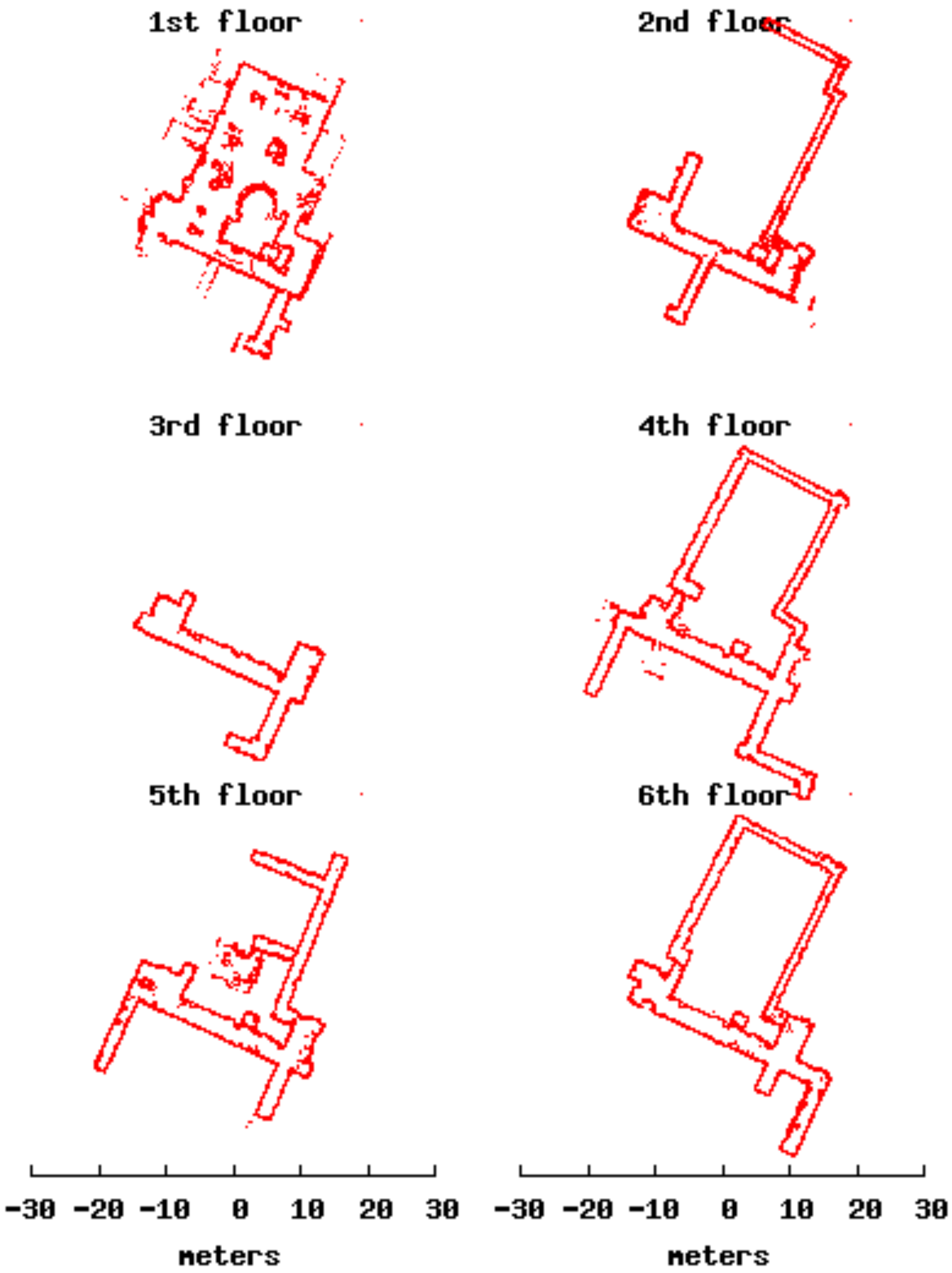

Figure 70, Individual maps of floors. These maps are segmented from the superimposed map in Figure 65, with the help of the barometric pressure log.

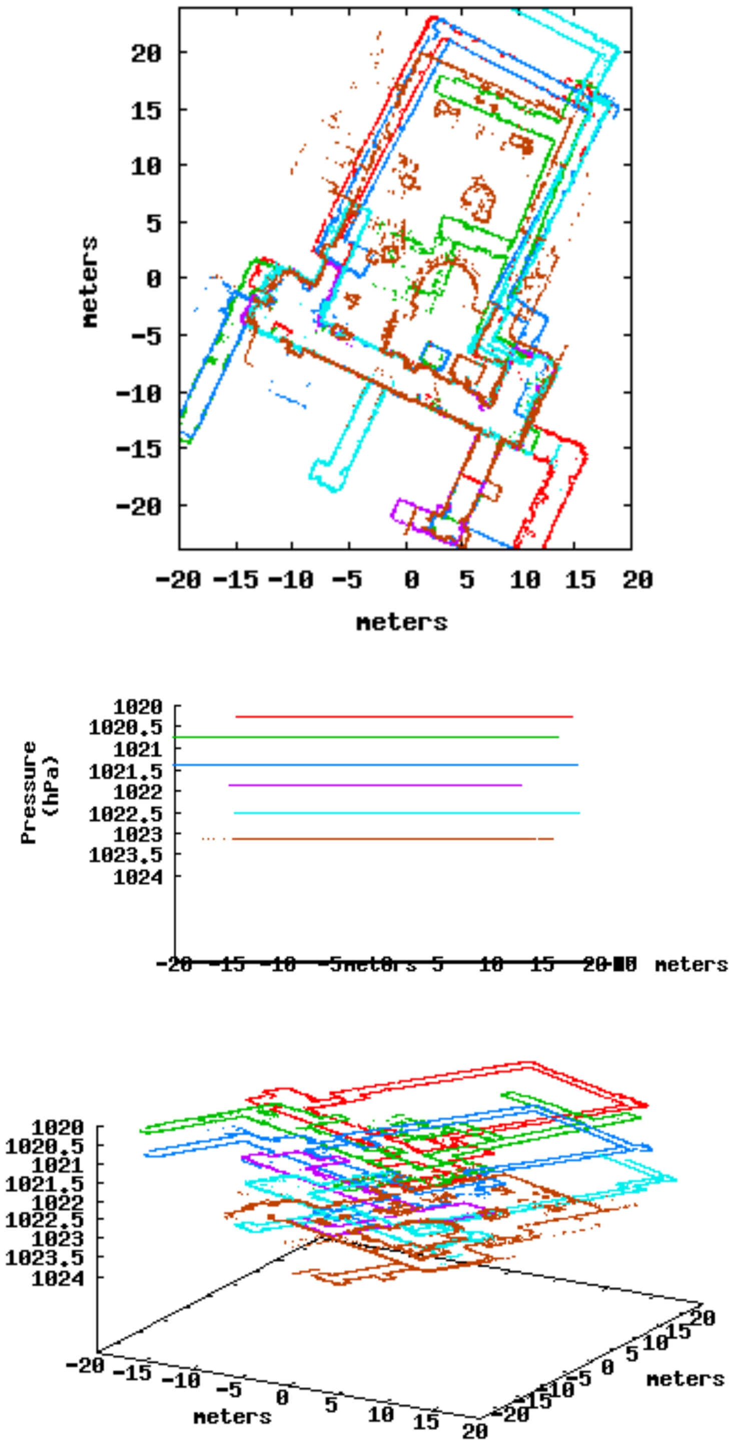

Figure 71, Resulting multi floor map. 2D floor maps are the result of metric SLAM (top). Elevations of individual floors are determined from barometric pressure (middle). The final representation is a 2.5D –sparse 3D- map of the building (bottom). Note the decreasing values of Pressure axis, in order to reflect the physical layout of the building.

Metric maps of the individual floors are simply generated by the 2D SLAM algorithm [122]. In general, the maps are consistent and aligned well with the other floors, especially around the main aisle, where the elevators are situated. Due to the shortcomings of the SLAM algorithms, slight misalignments are also detected, as in the map of the 2nd floor (Figure 70). Unlike the 4th floor and the 6th floor, a loop closure is not observed on the 2nd floor; and unlike the ground floor and the 5th floor, the environment is not rich in terms of features. A relatively featureless and long corridor without a loop closure results in poor self-localization of the robot, hence affects the generated floor map.

Figure 71 reveals that the elevations of the individual floors are referenced to the mean barometric pressures recorded at the corresponding floors during exploration. While the relative elevation differences can be represented using mean barometric pressures, they cannot be represented as absolute elevations from mean sea level because of the varying barometric pressure (as seen in Figure 67).

6.5.6 Conclusion and Future Work

In this paper, we presented a novel method for mapping of multi-floor buildings. It is based on laser range measurements to build metric maps of the environment, and barometric pressure measurements to detect floor transitions and segment the metric maps.

The method is tested with a real robot in a typical multi-floor building, and the results show that the method is capable of generating physically consistent maps. It is simple, efficient, and easily applicable to the existing robot systems with an addition of an inexpensive digital barometric sensor.

We aim to extend this framework by tracing the atmospheric pressure changes using stationary sensor nodes placed in the environment. This will allow us to maintain a globally referenced pressure map, and moreover enable mapping with multi-robot systems.

Bibliography

[1] A. G. Ozkil, S. Dawids, Z. Fan, and T. Sorensen, "Design of a robotic automation system for transportation of goods in hospitals," presented at the Computational Intelligence in Robotics and Automation, 2007. CIRA 2007. International Symposium on, 2007, pp. 392-397.

[2] "Aethon - TUG™: The Automated Robotic Delivery System." [Online]. Available: http://www.aethon.com/products/logistics.php. [Accessed: 01-Apr-2009].

[3] "AGV (Automated Guided Vehicle) Systems by FMC Technologies." [Online]. Available: http://www.fmcsgvs.com/content/products/atlis.htm. [Accessed: 01-Apr-2009].

[4] "Speciminder." [Online]. Available: http://www.ccsrobotics.com/products/speciminder.html. [Accessed: 01-Apr-2009].

[5] "Automatic Guided Vehicles for Healthcare." [Online]. Available: http://www.swisslog.com/index/hcs-index/hcs-systems/hcs-agv.htm. [Accessed: 01-Apr-2009].

[6] J. Carlson and R. R. Murphy, "Reliability analysis of mobile robots," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, 2003, vol. 1, p. 274–281.

[7] R. Brooks, "A robust layered control system for a mobile robot," *Robotics and Automation, IEEE Journal of*, vol. 2, no. 1, pp. 14-23, 1986.

[8] R. Brooks, "New Approaches to Robotics," *Science*, vol. 253, no. 5025, pp. 1227-1232, 1991.

[9] G. Engelberger, "HelpMate, a service robot with experience," *Industrial Robot: An International Journal*, vol. 25, no. 2, p. 101–104, 1998.

[10] *European Commission : CORDIS : FP7 : ICT : Projects : NESTOR : Robotic courier for on-demand transport of goods in hospitals and healthcare centres*. 2000.

[1] R. Murphy, *Introduction to AI Robotics*, MIT Press, 2000.

[12] N. J. Nilsson, *Shakey the Robot*. Storming Media, 1984.

[13] J. Engleberger, "Health-care robotics goes commercial: the HelpMate' experience," *Robotica(Cambridge. Print)*, vol. 11, pp. 517-523, 1993.

[14] J. Forlizzi and C. DiSalvo, "Service robots in the domestic environment: a study of the roomba vacuum in the home," in *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, 2006, p. 258–265.

[15] R. Arkin and T. Balch, "AuRA: principles and practice in review," Apr. 1997.

[16] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobilerobot navigation," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, 1991, pp. 1398-1404.

[17] J. K. Rosenblatt, "DAMN: a distributed architecture for mobile navigation," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 9, no. 2-3, pp. 339-360, 1997.

[18] K. Konolige and K. Myers, "The saphira architecture for autonomous mobile robots," 1998.

[19] R. R. Murphy and R. C. Arkin, "Sfx: An Architecture For Action-oriented Sensor Fusion," in *Intelligent Robots and Systems, 1992., Proceedings of the 1992 IEEE/RSJ International Conference on*, 1992, vol. 2.

[20] R. C. Arkin, *Behavior-Based Robotics*. MIT Press, 1998.

[21] J. Borenstein, H. R. Everett, and L. Feng, *Navigating Mobile Robots: Systems and Techniques*. AK Peters, Ltd. Natick, MA, USA, 1996.

[22] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. MIT Press, 2004.

[23] H. R. Everett, *Sensors for mobile robots: theory and application*. AK Peters, Ltd. Natick, MA, USA, 1995.

[24] "Energy and Transport - GALILEO." [Online]. Available: http://ec.europa.eu/dgs/energy_transport/galileo/index_en.htm. [Accessed: 10-Feb-2009].

[25] J. C. Harris, H. T. Inc, and V. A. Clifton, "An infogeometric approach to telerobotics," in *Telesystems Conference, 1994. Conference Proceedings., 1994 IEEE National*, 1994, pp. 153-156.

[26] A. M. Ladd, K. E. Bekris, A. Rudys, L. E. Kavraki, and D. S. Wallach, "Robotics-Based Location Sensing Using Wireless Ethernet," *Wireless Networks*, vol. 11, no. 1, pp. 189-204, 2005.

[27] K. Briechle and U. D. Hanebeck, "Localization of a mobile robot using relative bearing measurements," *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 1, pp. 36-44, 2004.

[28] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393-422, 2002.

[29] D. Hahnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose, "Mapping and localization with RFID technology," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, 2004, vol. 1.

[30] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Localization for mobile robot teams: A maximum likelihood approach," *USA: University of Sourthern California*, 2001.

[31] T. Goedemé, M. Nuttin, T. Tuytelaars, and L. Van Gool, "Omnidirectional Vision Based Topological Navigation," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 219-236, 2007.

[32] H. Surmann, K. Lingemann, A. Nuchter, and J. Hertzberg, "A 3D laser range finder for autonomous mobile robots," in *Proceedings of the 32nd ISR (International Symposium on Robotics)*, 2001, vol. 19, pp. 153-158.

[33] H. Surmann, A. Nüchter, and J. Hertzberg, "An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments," *Robotics and Autonomous Systems*, vol. 45, no. 3-4, pp. 181-198, 2003.

[34] C. Brenneke, O. Wulf, and B. Wagner, "Using 3D laser range data for SLAM in outdoor environments," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, 2003, vol. 1.

[35] S. Thrun, W. Burgard, and D. Fox, "A real-time algorithm for mobile robot mapping with applications tomulti-robot and 3D mapping," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, 2000, vol. 1.

[36] P. Buschka, *An Investigation of Hybrid Maps for Mobile Robots*. Örebro: Örebro universitetsbibliotek, 2005.

[37] P. Jensfelt, *Approaches to Mobile Robot Localization in Indoor Environments*. Stockholm: KTH, Signals, Sensors and Systems.

[38] J. Gasós and A. Martín, "A fuzzy approach to build sonar maps for mobile robots," *Computers in Industry*, vol. 32, no. 2, pp. 151-167, 1996.

[39] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobilerobot," in *Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on*, 1991, pp. 1442-1447.

[40] W. D. Rencken, "Autonomous sonar navigation in indoor, unknown and unstructuredenvironments," in *Intelligent Robots and Systems' 94.'Advanced Robotic Systems and the Real World', IROS'94. Proceedings of the IEEE/RSJ/GI International Conference on*, 1994, vol. 1.

[41] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, 1985, vol. 2.

[42] A. Saffiotti, "The uses of fuzzy logic in autonomous robot navigation: a catalogue raisonne," *Soft Computing*, vol. 1, no. 4, pp. 180-197, 1997.

[43] I. Kimura and J. Tadano, "Guide and carry robot for hospital use," *Advanced robotics*, vol. 3, no. 3, pp. 213-220, 1989.

[44] H. F. M. Van der Loos and D. J. Reinkensmeyer, "Rehabilitation and Health Care Robotics," in *Springer Handbook of Robotics*, 2008, pp. 1223-1251.

[45] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artificial Intelligence*, vol. 99, no. 1, pp. 21-71, 1998.

[46] E. Fabrizi and A. Saffiotti, "Extracting topology-based maps from gridmaps," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, 2000, vol. 3.

[47] T. Duckett, S. Marsland, and J. Shapiro, "Fast, On-Line Learning of Globally Consistent Maps," *Autonomous Robots*, vol. 12, no. 3, pp. 287-300, 2002.

[48] B. Kuipers, "Modeling spatial knowledge," *Cognitive Science*, vol. 2, no. 2, pp. 129-153, 1978.

[49] T. Duckett and A. Saffiotti, "Building globally consistent gridmaps from topologies," in *Proceedings of the Sixth International IFAC Symposium on Robot Control (SYROCO), Wien, Austria*, 2000.

[50] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. A. Fernandez-Madrigal, and J. Gonzalez, "Multi-hierarchical semantic maps for mobile robotics," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, 2005, pp. 2278-2283.

[51] J. A. Fernandez-Madrigal, "Assistive navigation of a robotic wheelchair using a multihierarchical model of the environment," *Integrated Computer-Aided Engineering*, vol. 11, no. 4, pp. 309-322, 2004.

[52] J. I. Nieto, J. E. Guivant, and E. M. Nebot, "The HYbrid metric maps (HYMMs): a novel map representation for DenseSLAM," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 1.

[53] B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli, "Local metrical and global topological maps in the hybrid spatial semantic hierarchy," in *2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04*, vol. 5.

[54] H. Shatkay and L. P. Kaelbling, "Learning topological maps with weak local odometric information," *In Proceedings of IJCAI-97. IJCAI, Inc*, 1997.

[55] J. Vannoy and J. Xiao, "Real-Time Adaptive Motion Planning (RAMP) of Mobile Manipulators in Dynamic Environments With Unforeseen Changes," *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 1199-1212, 2008.

[56] S. Thrun and A. Bücken, "Integrating grid-based and topological maps for mobile robot navigation," in *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, 1996, pp. 944-951.

[57] E. Aguirre and A. Gonzalez, "Integrating fuzzy topological maps and fuzzy geometric maps for behavior-based robots," *International Journal of Intelligent Systems*, vol. 17, no. 3, pp. 333-368, 2002.

[58] R. Chatila and J. Laumond, "Position referencing and consistent world modeling for mobile robots," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, 1985, vol. 2.

[59] S. Thrun, W. Burgard, and D. Fox, "A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots," *Autonomous Robots*, vol. 5, no. 3, pp. 253-271, 1998.

[60] T. Duckett, "A Genetic Algorithm for Simultaneous Localization and Mapping," in *IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION*, 2003, vol. 1, pp. 434-439.

[61] A. Eliazar and R. Parr, "DP-SLAM: Fast, Robust Simultaneous Localization and Mapping Without Predetermined Landmarks," in *INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, 2003, vol. 18, pp. 1135-1142.

[62] K. Murphy, "Bayesian map learning in dynamic environments," *Advances in Neural Information Processing Systems (NIPS)*, vol. 12, p. 1015–1021, 1999.

[63] G. Grisetti, G. D. Tipaldi, C. Stachniss, W. Burgard, and D. Nardi, "Fast and accurate SLAM with Rao–Blackwellized particle filters," *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 30-38, 2007.

[64] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters," *Robotics, IEEE Transactions on*, vol. 23, no. 1, pp. 34-46, 2007.

[65] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," *Autonomous Robot Vehicles*, vol. 1, pp. 167-193, 1990.

[66] U. Frese and G. Hirzinger, "Simultaneous localization and mapping-a discussion," in *Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics*, 2001, p. 17–26.

[67] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and A. Y. Ng, *Simultaneous Mapping and Localization with Sparse Extended Information Filters: Theory and Initial Results*. Springer, 2002.

[68] K. S. Chong and L. Kleeman, "Mobile robot map building from an advanced sonar array and accurate odometry," *International Journal of Robotics Research*, vol. 18, no. 1, pp. 20-36, 1999.
[69] R. Martinez-Cantin and J. A. Castellanos, "Unscented SLAM for Large-Scale Outdoor Environments," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2005, p. 328–333.
[70] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, p. 35–45, 1960.
[71] J. P. Trevelyan, S.-C. Kang, and W. R. Hamel, "Robotics in Hazardous Applications," in *Springer Handbook of Robotics*, 2008, pp. 1101-1126.
[72] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376-382, 1991.
[73] J. Gasós and A. Saffiotti, "Using fuzzy sets to represent uncertain spatial knowledge in autonomous robots," *Spatial Cognition and Computation*, vol. 1, no. 3, pp. 205-226, 1999.
[74] P. Jensfelt, "Localization using laser scanning and minimalistic environmental models," *Licentiate thesis, Automatic Control, Royal Institute of Technology, SE-100*, vol. 44, 1999.
[75] A. Saffiotti and L. P. Wesley, "Perception-based self-localization using fuzzy locations," *Lecture Notes in Computer Science*, pp. 368-384, 1996.
[76] D. Fox, W. Burgard, and S. Thrun, "Active markov localization for mobile robots," *Robotics and Autonomous Systems*, vol. 25, no. 3, pp. 195-208, 1998.
[77] W. Burgard, A. Derr, D. Fox, and A. B. Cremers, "Integrating global position estimation and position tracking formobile robots: the dynamic Markov localization approach," in *1998 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1998. Proceedings.*, 1998, vol. 2.
[78] P. Buschka, A. Saffiotti, and Z. Wasik, "Fuzzy landmark-based localization for a legged robot," in *2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings*, 2000, vol. 2.
[79] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *1999 IEEE International Conference on Robotics and Automation, 1999. Proceedings*, 1999, vol. 2.
[80] D. Fox, "KLD-sampling: Adaptive particle filters and mobile robot localization," *Advances in Neural Information Processing Systems (NIPS)*, 2001.
[81] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *To appear in Artificial Intelligence*, 2001.
[82] P. Jensfelt and T. högskolan i Stockholm, *Approaches to mobile robot localization in indoor environments*. Tekniska högsk., 2001.
[83] J. de Lope and D. Maravall, "Integration of reactive utilitarian navigation and topological modeling," 2003.
[84] U. Nehmzow and C. Owen, "Robot navigation in the real world: Experiments with Manchester's FortyTwo in unmodified, large environments," *Robotics and Autonomous systems*, vol. 33, no. 4, pp. 223-242, 2000.
[85] C. H. Choi, J. B. Song, W. Chung, and M. Kim, "Topological map building based on thinning and its application to localization," in *IEEE/RSJ International Conference on Intelligent Robots and System, 2002*, 2002, vol. 1.

[86] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," in *IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA'00*, 2000, vol. 2.
[87] J. C. Latombe, *Robot motion planning*. Kluwer academic publishers, 1991.
[88] N. J. Nilsson, *A mobile automaton: An application of artificial intelligence techniques*. Storming Media, 1969.
[89] H. Choset and J. Burdick, "Sensor-based exploration: The hierarchical generalized voronoi graph," *The International Journal of Robotics Research*, vol. 19, no. 2, p. 96, 2000.
[90] C. Freksa, R. Moratz, and T. Barkowsky, "Schematic maps for robot navigation," *Lecture notes in computer science*, pp. 100-114, 2000.
[91] H. Choset, "Coverage for robotics – A survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1, pp. 113-126, Oct. 2001.
[92] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, p. 90, 1986.
[93] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100-107, 1968.
[94] A. Stentz, "Optimal and Efficient Path Planning for Partially-Known Environments."
[95] S. Thrun, J.-S. Gutmann, D. Fox, W. Burgard, and B. Kuipers, "Integrating Topological and Metric Maps for Mobile Robot Navigation: A Statistical Approach," in *AAAI/IAAI*, 1998, pp. 989-995.
[96] C. Galindo, J. A. Fernandez-Madrigal, and J. Gonzalez, "Improving efficiency in mobile robot task planning through world abstraction," *IEEE Transactions on Robotics*, vol. 20, no. 4, pp. 677-690, 2004.
[97] J. A. Fernández-Madrigal, "Assistive navigation of a robotic wheelchair using a multihierarchical model of the environment," *Integrated Computer-Aided Engineering*, vol. 11, no. 4, pp. 309-322, 2004.
[98] I. Ulrich and J. Borenstein, "VFH+: reliable obstacle avoidance for fast mobile robots," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, 1998, vol. 2.
[99] I. Ulrich and J. Borenstein, "VFH*: local obstacle avoidance with look-aheadverification," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, 2000, vol. 3.
[100] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *Robotics & Automation Magazine, IEEE*, vol. 4, no. 1, pp. 23-33, 1997.
[101] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, 1999, vol. 1, pp. 341-346 vol.1.
[102] J. Minguez and L. Montano, "Nearness Diagram Navigation (ND): A New Real Time Collision Avoidance Approach," *In Proc. of the ieee/rsj international conference on intelligent robots and systems (iros'00*, vol. 60, p. 2094--2100, 2000.
[103] J. Minguez and L. Montano, "Nearness Diagram (ND) Navigation: Collision Avoidance in Troublesome Scenarios," *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, vol. 20, no. 1, p. 45, 2004.

[104] J. Jackson, "Microsoft robotics studio: A technical introduction," *Robotics & Automation Magazine, IEEE*, vol. 14, no. 4, pp. 82-87, 2007.

[105] B. P. Gerkey, R. T. Vaughan, K. Stoy, A. Howard, G. S. Sukhatme, and M. J. Mataric, "Most valuable player: a robot device server for distributed control," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, 2001, vol. 3, pp. 1226-1231 vol.3.

[106] T. M. P. and I. R. L. University of Malaga, "The Mobile Robot Programming Toolkit (MRPT)," *The Mobile Robot Programming Toolkit (MRPT)*. [Online]. Available: http://babel.isa.uma.es/mrpt/index.php/Main_Page. [Accessed: 15-Sep-2009].

[107] M. Montemerlo, N. Roy, and S. Thrun, "Perspectives on standardization in mobile robot programming: The Carnegie Mellon navigation (CARMEN) toolkit," in *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings*, 2003, vol. 3.

[108] MobileRobots inc., "P3dx General purpose robot base," *P3dx General purpose robot base*. [Online]. Available: http://www.activrobots.com/ROBOTS/p2dx.html. [Accessed: 15-Sep-2009].

[109] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, p. 1052, 2007.

[110] D. Wagner and D. Schmalstieg, "Artoolkitplus for pose tracking on mobile devices," in *Computer Vision Winter Workshop*, 2007, pp. 6-8.

[111] I. Nielsen, "Analysis of the tunnel usage in Bispebjerg Hospital," 10-Jan-2011. [Online]. Available: http://www.bispebjerghospital.dk/menu/Om+Bispebjerg+Hospital/nybyggeri-paa-bispebjerg/Aktuelt_nybyggeri/tunnelanalyse.htm. [Accessed: 19-Apr-2011].

[112] P. Team, *Pov-ray—The persistence of vision ray tracer*. 1998.

[113] D. G. Lowe, "Object recognition from local scale-invariant features," in *iccv*, 1999, p. 1150.

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

[115] A. Gil, O. Reinoso, A. Vicente, C. Fernández, and L. Payá, "Monte Carlo localization using SIFT features," *Pattern Recognition and Image Analysis*, p. 623–630, 2005.

[116] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1052–1067, 2007.

[117] K. Konolige and M. Agrawal, "Frameslam: From bundle adjustment to real-time visual mapping."

[118] C. Wu, *SiftGPU: A GPU implementation of David Lowe's scale invariant feature transform (SIFT)*. .

[119] S. Warn, W. Emeneker, J. Cothren, and A. Apon, "Accelerating SIFT on Parallel Architectures."

[120] S. Heymann, K. Maller, A. Smolic, B. Froehlich, and T. Wiegand, "SIFT implementation and optimization for general-purpose GPU," in *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2007.

[121] S. N. Sinha, J. M. Frahm, M. Pollefeys, and Y. Genc, “GPU-based video feature tracking and matching,” in *EDGE, Workshop on Edge Computing Using New Commodity Architectures*, 2006, vol. 278.

[122] G. GrissettiW. Burgard, “Gmapping,” *http://www.openslam.org/gmapping.html*. [Online]. Available: http://www.openslam.org/gmapping.html.

[123] E. Poulin, “Benchmarking the hospital logistics process A potential cure for the ailing health care sector,” *CMA MANAGEMENT*, vol. 77, no. 1, pp. 20-23, 2003.

[124] “Danmarks Statistikbank, Social conditions, health and justice, table UD2.” [Online]. Available: http://www.statistikbanken.dk/UD2. [Accessed: 01-Apr-2009].

[125] “Health services - Statistics by subject, 03.02.” [Online]. Available: http://www.ssb.no/english/subjects/03/02/. [Accessed: 01-Apr-2009].

[126] B. Graf, M. Hans, and R. D. Schraft, “Care-O-bot II—Development of a next generation robotic home assistant,” *Autonomous robots*, vol. 16, no. 2, pp. 193-205, 2004.

[127] F. Carreira, T. Canas, A. Silva, and C. Cardeira, “i-Merc: A Mobile Robot to Deliver Meals inside Health Services,” in *Proceedings of the 2006 IEEE Internacional Conference on Cibernetics & Intelligence Systems & Robotics, Automation & Mechtronics*, 2006, pp. 1-8.

[128] T. Srensen, with A. G. Ozkil and S. Dawids, Zhun Fan, “Design of a robotic automation system for transportation of goods in hospitals,” in *Computational Intelligence in Robotics and Automation, 2007. CIRA 2007. International Symposium on*, 2007, pp. 392-397.

[129] Wikipedia contributors, “Barcode,” *Wikipedia, The Free Encyclopedia*. Wikimedia Foundation, 10-Jun-2009.

[130] “Domestic Mail Manual 708.4 - Special Standards, Technical Specifications, Barcoding Standards for Letters and Flats.” United States Postal Service.

[131] Denso-Wave Corp., “QR Code features | QR Code.com,” *http://www.denso-wave.com/qrcode/qrfeature-e.html*, 15-Jun-2009. [Online]. Available: http://www.denso-wave.com/qrcode/qrfeature-e.html. [Accessed: 14-Jun-2009].

[132] ISO - International Organization for Standardization, *ISO/IEC 18004:2000, Information technology – Automatic identification and data capture techniques – Bar code symbology – QR Code*. .

[133] I. S. Reed and G. Solomon, “Polynomial codes over certain finite fields,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, p. 300–304, 1960.

[134] ISO - International Organization for Standardization, *ISO/IEC 16022:2006, Information technology – Automatic identification and data capture techniques – Data Matrix bar code symbology specification*. .

[135] D. L. de Ipiña, P. R. S. Mendonça, and A. Hopper, “TRIP: A Low-Cost Vision-Based Location System for Ubiquitous Computing,” *Personal Ubiquitous Comput.*, vol. 6, no. 3, pp. 206-219, 2002.

[136] G. Jancke, “High Capacity Color Barcode Technology - Microsoft Research,” *http://research.microsoft.com/en-us/projects/hccb/about.aspx#robust*, 15-Jun-2009. .

[137] R. T. Azuma, others, “A survey of augmented reality,” *Presence-Teleoperators and Virtual Environments*, vol. 6, no. 4, p. 355–385, 1997.

[138] D. Stricker and T. Kettenbach, "Real-time and markerless vision-based tracking for outdoor augmented reality applications," in *Augmented Reality, 2001. Proceedings. IEEE and ACM International Symposium on*, 2001, pp. 189-190.
[139] V. Ferrari, T. Tuytelaars, and L. Van Gool, "Markerless augmented reality with a real-time affine region tracker," in *IEEE and ACM International Symposium on Augmented Reality, 2001. Proceedings*, 2001, pp. 87-96.
[140] Y. Genc, S. Riedel, F. Souvannavong, C. Akinlar, and N. Navab, "Marker-less tracking for AR: A learning-based approach," in *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, 2002.
[141] C. Yuan, "Markerless pose tracking for augmented reality," *Lecture Notes in Computer Science*, vol. 4291, p. 721, 2006.
[142] A. I. Comport, É. Marchand, and F. Chaumette, "A real-time tracker for markerless augmented reality," in *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, 2003.
[143] X. Zhang, S. Fronz, and N. Navab, "Visual marker detection and decoding in ar systems: A comparative study," in *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, 2002.
[144] H. Kato and M. Billinghurst, "Marker tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System," in *2nd IEEE and ACM International Workshop on Augmented Reality, 1999.(IWAR'99) Proceedings*, 1999, pp. 85-94.
[145] M. Fiala, "Vision guided control of multiple robots," in *Computer and Robot Vision, 2004. Proceedings. First Canadian Conference on*, 2004, pp. 241-246.
[146] R. Bencina and M. Kaltenbrunner, "The Design and Evolution of Fiducials for the reacTIVision System," in *Proc. 3rd International Conference on Generative Systems in the Electronic Arts*, 2005.
[147] E. Costanza and J. Robinson, "A Region Adjacency Tree Approach to the Detection and Design of Fiducials," presented at the Video, Vision and Graphics, York, United Kingdom, 2003.
[148] G. N. Desouza and A. C. Kak, "Vision for mobile robot navigation: a survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 2, pp. 237-267, 2002.
[149] T. Tuytelaars and K. Mikolajczyk, "Survey on local invariant features," *FnT Computer Graphics and Vision*, vol. 1, no. 1, pp. 1-94, 2008.
[150] M. Kabuka and A. Arenas, "Position verification of a mobile robot using standard pattern," *Robotics and Automation, IEEE Journal of*, vol. 3, no. 6, pp. 505-516, 1987.
[151] C. Becker, J. Salas, K. Tokusei, and J.-C. Latombe, "Reliable navigation using landmarks," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, 1995, vol. 1, pp. 401-406 vol.1.
[152] C.-C. Lin and R. L. Tummala, "Mobile robot navigation using artificial landmarks," *Journal of Robotic Systems*, vol. 14, no. 2, pp. 93-106, 1997.
[153] C. J. Taylor and D. J. Kriegman, "Vision-based motion planning and exploration algorithms for mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 3, pp. 417-426, 1998.
[154] K. Tashiro, J. Ota, Y. C. Lin, and T. Arai, "Design of the optimal arrangement of artificial landmarks," in *1995 IEEE International Conference on Robotics and Automation, 1995. Proceedings.*, 1995, vol. 1.

[155] G. Jang, S. Lee, and I. Kweon, "Color landmark based self-localization for indoor mobile robots," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 1, p. 1037–1042.
[156] M. Isard and A. Blake, "CONDENSATION—Conditional Density Propagation for Visual Tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5-28, 1998.
[157] I. R. Nourbakhsh, J. Bobenage, S. Grange, R. Lutz, R. Meyer, and A. Soto, "An affective mobile robot educator with a full-time job," *Artificial Intelligence*, vol. 114, no. 1-2, pp. 95-124, 1999.
[158] I. R. Nourbakhsh, C. Kunz, and T. Willeke, "The mobot museum robot installations: a five year experiment," presented at the Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on, 2003, vol. 4, pp. 3636-3641 vol.3.
[159] D. Scharstein and A. Briggs, "Fast recognition of self-similar landmarks," 1999, vol. 26.
[160] A. J. Briggs, D. Scharstein, D. Braziunas, C. Dima, and P. Wall, "Mobile robot navigation using self-similar landmarks," presented at the Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on, 2000, vol. 2, pp. 1428-1434 vol.2.
[161] D. Scharstein and A. J. Briggs, "Real-time recognition of self-similar landmarks," *Image and Vision Computing*, vol. 19, no. 11, pp. 763-772, 2001.
[162] C. Pradalier, "Vision-Based Handling Tasks for an Autonomous Outdoor Forklift," in *Field and Service Robotics*, 2008, pp. 61-70.
[163] E. Celaya, J. L. Albarral, P. Jimenez, and C. Torras, "Visually-Guided Robot Navigation: From Artificial To Natural Landmarks," 2008, vol. 42, pp. 287-296.
[164] J. Sattar, E. Bourque, P. Giguere, and G. Dudek, "Fourier tags: Smoothly degradable fiducial markers for use in human-robot interaction," presented at the Computer and Robot Vision, 2007. CRV '07. Fourth Canadian Conference on, 2007, pp. 165-174.
[165] M. Fiala, "Linear markers for robot navigation with panoramic vision," presented at the Computer and Robot Vision, 2004. Proceedings. First Canadian Conference on, 2004, pp. 145-154.
[166] M. Fiala, "Vision guided control of multiple robots," presented at the Computer and Robot Vision, 2004. Proceedings. First Canadian Conference on, 2004, pp. 241-246.
[167] O. Orqueda and R. Fierro, *A visual tracking system for controlling mobile robots in formation*. Dec, 2006.
[168] X. Li, G. Tian, H. Xue, F. Lu, and P. Zhang, "Study on Object Operating System for Service Robot Based on Artificial Marks," 2007, pp. 603-607.
[169] J. Huh, W. S. Chung, S. Y. Nam, and W. K. Chung, "Mobile Robot Exploration in Indoor Environment Using Topological Structure with Invisible Barcodes," *ETRI Journal*, vol. 29, no. 2, pp. 189-200, 2007.
[170] M. Fiala, "ARTAG Rev2 Fiducial Marker System: Vision based Tracking for AR," in *Workshop of Industrial Augmented Reality*, 2005.
[171] D. W. Zeitler, A. R. Black, and C. M. A. Ko, *Automated guided vehicle (AGV) with bipolar magnet sensing*. Google Patents, 2002.

[172] H. Northfield and A. U. H. Sheikh, "An autonomous vehicle with fuzzy logic navigational control," in *Vehicle Navigation and Information Systems Conference, 1993., Proceedings of the IEEE-IEE*, 1993, p. 486–489.
[173] U. Wiklund, U. Andersson, and K. Hyyppa, "AGV navigation by angle measurements," in *Automated Guided Vehicle Systems: Proceedings of the 6th International Conference: 25-26 October 1988, Brussels, Belgium*, 1988, vol. 199, p. 199.
[174] T. Kampke, B. Kluge, E. Prassler, and M. Strobel, "Robot Position Estimation on a RFID-Tagged Smart Floor," in *Field and Service Robotics: Results of the 6 th International Conference(STAR: Springer Tracts in Advanced Robotics Series Volume 42)*, 2008, vol. 42, p. 201–211.
[175] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," *AI Magazine*, vol. 29, no. 1, p. 9–20, 2008.
[176] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, p. 46–57, 1989.
[177] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): Towards exact localization without explicit localization," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 2, pp. 125-137, 2001.
[178] A. Elfes, "A sonar-based mapping and navigation system," in *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, 1986, vol. 3.
[179] S. D. Jones, C. Andersen, and J. L. Crowley, "Appearance based processes for visual navigation," in *Proc. IEEE Int'l Conf. Intelligent Robots and Systems*, 1997, p. 551–557.
[180] D. Filliat and J. A. Meyer, "Map-based navigation in mobile robots: I. A review of localization strategies," *Cognitive Systems Research*, vol. 4, no. 4, pp. 243-282, 2003.
[181] N. Tomatis, "Hybrid, Metric-Topological Representation for Localization and Mapping," in *Robotics and Cognitive Approaches to Spatial Mapping*, 2008, pp. 43-63.
[182] B. Lisien, D. Morales, D. Silver, G. Kantor, I. Rekleitis, and H. Choset, "The hierarchical atlas," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 473-481, 2005.
[183] The Robotics Institute of CMU, "CARMEN Robot Navigation Toolkit," *http://carmen.sourceforge.net/home.html.* [Online]. Available: http://carmen.sourceforge.net/home.html. [Accessed: 15-Sep-2009].
[184] A. G. Ozkil, Z. Fan, S. Dawids, J. Klæstrup Kristensen, K. H. Christensen, and H. Aanæs, "Mobile Robot Navigation Using Visual Tags : A Review," in *Proceedings of Robotics and Applications, IASTED-RA 2009*, 2009.
[185] A. G. Ozkil, Z. Fan, J. Xiao, J. K. Kristensen, K. H. Christensen, and H. Aanæs, "Empirical Evaluation of a Practical Indoor Mobile Robot Navigation Method Using Hybrid Maps," in *Submitted to: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.
[186] A. G. Ozkil, Z. Fan, S. Dawids, J. Klæstrup Kristensen, K. H. Christensen, and H. Aanæs, "Service Robots for Hospitals: A Case Study of Transportation Tasks in a Hospital," in *Proceesings of 2009 IEEE International Conference on Automation and Logistics*, 2009, pp. 289 - 294.

[187] J. Hightower, R. Want, and G. Borriello, “SpotON: An indoor 3D location sensing technology based on RF signal strength,” *UW CSE 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA*, 2000.

[188] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, “LANDMARC: indoor location sensing using active RFID,” *Wireless Networks*, vol. 10, no. 6, p. 701–710, 2004.

[189] H. H. Kim, K. N. Ha, S. Lee, and K. C. Lee, “Resident location-recognition algorithm using a Bayesian classifier in the PIR sensor-based indoor location-aware system,” *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 39, no. 2, p. 240–245, 2009.

[190] H. Liu, H. Darabi, P. Banerjee, and J. Liu, “Survey of wireless indoor positioning techniques and systems,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 6, p. 1067–1080, 2007.

[191] ISO - International Organization for Standardization, *ISO/IEC 15420:2000, Information technology – Automatic identification and data capture techniques – Bar code symbology specification – EAN/UPC*. .

[192] ISO - International Organization for Standardization, “ISO/IEC 16388:2007 - Information technology -- Automatic identification and data capture techniques -- Code 39 bar code symbology specification.” [Online]. Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=43897. [Accessed: 25-Nov-2010].

[193] ISO - International Organization for Standardization, *ISO/IEC 18004:2000, Information technology – Automatic identification and data capture techniques – Bar code symbology – QR Code*. .

[194] ISO - International Organization for Standardization, *ISO/IEC 16022:2006, Information technology – Automatic identification and data capture techniques – Data Matrix bar code symbology specification*. .

[195] C. Plain-Jones and J. E. Ludlow, “Machine vision in the electronics industry,” *Assembly Automation*, vol. 15, no. 2, p. 30–35, 1995.

[196] M. Rohs and B. Gfeller, “Using camera-equipped mobile phones for interacting with real-world objects,” *Advances in Pervasive Computing*, p. 265–271, 2004.

[197] G. Jancke, “High Capacity Color Barcode Technology - Microsoft Research,” *http://research.microsoft.com/en-us/projects/hccb/about.aspx#robust*, 15-Jun-2009. .

[198] N. Navab, “Industrial augmented reality (IAR): challenges in design and commercialization of killer apps,” in *Mixed and Augmented Reality, 2003. Proceedings. The Second IEEE and ACM International Symposium on*, 2003, pp. 2-6.

[199] M. Kaltenbrunner and R. Bencina, “reacTIVision: a computer-vision framework for table-based tangible interaction,” in *Proceedings of the 1st international conference on Tangible and embedded interaction*, 2007, pp. 69-74.

[200] H. Kato and K. T. Tan, “Pervasive 2D Barcodes for Camera Phone Applications,” *Pervasive Computing, IEEE*, vol. 6, no. 4, pp. 76-85, 2007.

[201] *Comparison of Data Matrix and QR Code*. Consumer Electronics Association, 2001.

[202] *Choosing the best 2D barcode format for mobile apps*. Semacode Technical White Paper, 2006.
[203] M. Beetz et al., "Integrated, plan-based control of autonomous robots in human environments," *IEEE Intelligent Systems*, p. 56–65, 2001.
[204] S. Thrun et al., others, "MINERVA: A second generation mobile tour-guide robot," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999, vol. 2005.
[205] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. A. Fernandez-Madrigal, and J. Gonzalez, "Multi-hierarchical semantic maps for mobile robotics," 2005, pp. 2278-2283.
[206] M. Skubic et al., "Spatial language for human-robot dialogs," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 34, no. 2, p. 154–167, 2004.
[207] C. Theobalt et al., "Talking to Godot: Dialogue with a mobile robot," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, 2002, p. 1338–1343.
[208] S. Vasudevan, S. G\ächter, V. Nguyen, and R. Siegwart, "Cognitive maps for mobile robots–an object based approach," *Robotics and Autonomous Systems*, vol. 55, no. 5, p. 359–371, 2007.
[209] M. Rohs, "PhD Thesis: Linking Physical and Virtual Worlds with Visual Markers and Handheld Devices," ETH Zurich, 2005.
[210] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, p. 155–166, 2009.
[211] D. Harwood, with Yubin Hung and Pen-Shu Yeh, "Passive ranging to known planar point sets," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, 1985, vol. 2, pp. 80-85.
[212] "libdmtx Home," *Open source software for reading and writing Data Matrix 2D barcodes, http://www.libdmtx.org/*. [Online]. Available: http://www.libdmtx.org/. [Accessed: 09-Aug-2010].
[213] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *American Control Conference, 1995. Proceedings of the*, 1995, p. 1628–1632.
[214] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using FastSLAM," in *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA'03*, 2003, vol. 2.
[215] F. Lu and E. Milios, "Globally Consistent Range Scan Alignment for Environment Mapping," *Auton. Robots*, vol. 4, no. 4, pp. 333-349, 1997.
[216] L. Iocchi and S. Pellegrini, "Building 3d maps with semantic elements integrating 2d laser, stereo vision and imu on a mobile robot," in *IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR'07)*, 2007.
[217] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *The International Journal of Robotics Research*, vol. 25, no. 12, p. 1243, 2006.
[218] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai, "A practical, decision-theoretic approach to multi-robot mapping and exploration," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS*, 2003, p. 3232–3238.

[219] M. Karg, K. M. Wurm, C. Stachniss, K. Dietmayer, and W. Burgard, "Consistent Mapping of Multistory Buildings by Introducing Global Constraints to Graph-based SLAM," presented at the In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), Anchorage, Alaska, 2010.

[220] N. Berberan-Santos, E. N. Bodunov, and L. Pogliani, "On the barometric formula," *American Journal of Physics*, vol. 65, no. 5, p. 404–412, 1997.

[221] ISO - International Organization for Standardization, "ISO 2533:1975, Standard Atmosphere," 1975. .

[222] D. C. Montgomery and L. A. Johnson, *Forecasting and time series analysis*. McGraw-Hill Companies, 1976.

[223] C. Croarkin, P. Tobias, and C. Zey, *Engineering statistics handbook*. The National Institute of Standards and Technology (US), 2001.

Hospitals are complex and dynamic organisms that are vital to the well-being of societies. Providing good quality healthcare is the ultimate goal of a hospital, and it is what most of us are only concerned with. A hospital, on the other hand, has to orchestrate a great deal of supplementary services to maintain the quality of healthcare provided.

This thesis and the Industrial PhD project aim to address logistics, which is the most resource demanding service in a hospital. The scale of the transportation tasks is huge and the material flow in a hospital is comparable to that of a factory.

We believe that these transportation tasks, to a great extent, can be and will be automated using mobile robots. This thesis consequently addresses the key technical issues of implementing service robots in hospitals.

ISBN 978-87-92706-23-2

DTU Management Engineering
Department of Management Engineering
Technical University of Denmark

Produktionstorvet
Building 424
DK-2800 Kongens Lyngby
Denmark
Tel. +45 45 25 48 00
Fax +45 45 93 34 35

www.man.dtu.dk