INFORMATION-THEORETIC ENVIRONMENT MODELING FOR MOBILE ROBOT LOCALIZATION

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der naturwissenschaften
der Universität Mannheim

vorgelegt von

Sherine Rady Abdel Ghany Elasmar
– aus Ägypten, Kairo –

Mannheim, January 2012

Dekan: Prof. Dr. H. J. Müller, Universität Mannheim
Referent: Prof. Dr. E. Badreddin, Universität Heidelberg
Korreferent: Prof. Dr. R. Männer, Universität Heidelberg

Tag der mündlichen Prüfung: 28. Juni 2012

Abstract

To enhance robotic computational efficiency without degenerating accuracy, it is imperative to fit the right and exact amount of information in its simplest form to the investigated task. This thesis conforms to this reasoning in environment model building and robot localization. It puts forth an approach towards building maps and localizing a mobile robot efficiently with respect to unknown, unstructured and moderately dynamic environments. For this, the environment is modeled on an information-theoretic basis, more specifically in terms of its transmission property. Subsequently, the presented environment model, which does not specifically adhere to classical geometric modeling, succeeds in solving the environment disambiguation effectively.

The proposed solution lays out a two-level hierarchical structure for localization. The structure makes use of extracted features, which are stored in two different resolutions in a single hybrid feature-map. This enables dual coarse-topological and fine-geometric localization modalities.

The first level in the hierarchy describes the environment topologically, where a defined set of places is described by a probabilistic feature representation. A conditional entropy-based criterion is proposed to quantify the transinformation between the feature and the place domains. This criterion provides a double benefit of pruning the large dimensional feature space, and at the same time selecting the best discriminative features that overcome environment aliasing problems. Features with the highest transinformation are filtered and compressed to form a coarse resolution feature-map (codebook). Localization at this level is conducted through place matching.

In the second level of the hierarchy, the map is viewed in high-resolution, as consisting of non-compressed entropy-processed features. These features are additionally tagged with their position information. Given the identified topological place provided by the first level, fine localization corresponding to the second level is executed using feature triangulation. To enhance the triangulation accuracy, redundant features are used and two metric evaluating criteria are employed; one for dynamic features and mismatches detection, and another for feature selection.

The proposed approach and methods have been tested in realistic indoor environments using a vision sensor and the Scale Invariant Feature Transform local feature extraction. Through experiments, it is demonstrated that an information-theoretic modeling approach is highly efficient in attaining combined accuracy and computational efficiency performances for localization. It has also been proven that the approach is capable of modeling environments with a high degree of unstructuredness, perceptual aliasing, and dynamic variations (illumination conditions; scene dynamics). The merit of employing this modeling type is that environment features are evaluated quantitatively, while at the same time qualitative conclusions are generated about feature selection and performance in a robot localization task. In this way, the accuracy of localization can be adapted in accordance with the available resources.

The experimental results also show that the hybrid topological-metric map provides sufficient information to localize a mobile robot on two scales, independent of the robot motion model. The codebook exhibits fast and accurate topological localization at significant compression ratios. The hierarchical localization framework demonstrates robustness and optimized space and time complexities. This, in turn, provides scalability to large environments application and real-time employment adequacies.

Keywords:

Mobile robots, information theory, transmission, transinformation, environment modeling, map building, hybrid map, localization, hierarchical localization, place recognition, triangulation, feature evaluation and selection, multi-resolution features, entropy-based features, codewords, codebook, local features, Scale Invariant Feature Transform, perceptual aliasing, correspondence problem, dynamic environment, unstructured environment, complexity, scalability.

Zusammenfassung

Zur Erhöhung der Recheneffizienz bei der Steuerung autonomer mobiler Roboter ist es erforderlich die Menge der übermittelten Information sowie deren Repräsentation genau an die Aufgabe anzupassen. Diese Dissertation behandelt diese Fragestellung in Bezug auf die Umweltmodellierung und die Lokalisierung von mobilen Robotern, wobei für die Erzeugung von Umgebungskarten und die darauf basierende Lokalisation von unbekannten unstrukturierten und moderat veränderlichen Umgebungsbedingungen ausgegangen wird. Im Gegensatz zu klassischen Geometrie-basierten Verfahren, wird hier die Umgebung auf infomationstheoretischen Grundlagen insbesondere unter Berücksichtigung von Transmissionseigenschaften modelliert und zur Lösung des Unterscheidbarkeitsproblems verwendet.

Die vorgeschlagene Lösung besitzt eine hierarchische Struktur mit zwei Ebenen in der extrahierte Umgebungsmerkmale in zwei unterschiedlichen Auflösungen in Form einer einheitlichen hybriden Karte abgelegt werden. Dies ermöglicht sowohl eine topologische Groblokalisierung als auch eine geometrische Feinlokalisierung.

Die erste Hierarchieebene beschreibt die Topologie der Umgebung, in der eine Menge gegebener Orte durch eine probabilistische Merkmalsrepräsentation definiert werden. Ein Kriterium wird vorgeschlagen, das auf der bedingten Entropie zur Beschreibung der Transinformation zwischen Merkmals- und Ortsbereich beruht. Dieses Kriterium besitzt den Vorteil, den hochdimensionalen Merkmalsraum zu reduzieren und gleichzeitig die Merkmale mit der höchsten Unterscheidbarkeit auszuwählen, die zur eindeutigen Ortswahrnehmung dienen. Merkmale mit höherer Transinformation werden gefiltert und zur Erzeugung einer grob aufgelösten Merkmalskarte (Code-Tabelle) komprimiert. Die Orterkennung auf dieser Ebene geschieht durch einen Vergleich der Sensorwerte mit den Datenbankeinträgen.

Auf der zweiten Ebene der Hierarchie wird die Umgebungskarte in hochauflösender Form mit unkomprimierten Merkmalen dargestellt, die in Bezug auf deren Entropie verarbeitet wurden. Die Merkmale sind zusätzlich mit der entsprechenden Positionsinformation belegt. Mit dem durch die erste Ebene gegebenen Ort wird die Feinpositionierung entsprechend der zweiten Ebenen mit Hilfe einer Triangulation durchgeführt. Um die Genauigkeit der Triangulation zu erhöhen, werden redundante Merkmale verwendet und zwei Kriterien angewandt: Das erste zur Erkennung von Merkmalsveränderungen (z.B. Verschiebungen in der Szene) und von Nicht-Übereinstimmung, das zweite zur Merkmalsauswahl.

Die vorgeschlagenen Ansätze und Methoden wurden in einer realistischen Innenraum-Umgebung mit einem Bildsensor und der sogenannten SIFT- (eng. Scale Invariant Feature Transform) Extraktion getestet. Experimente haben gezeigt, dass der informationstheoretische Modellierungsansatz hoch effizient in Bezug auf Genauigkeit und Ausnutzung der Rechenleistung ist. Weiterhin wurde gezeigt, dass der Ansatz geeignet ist, Umgebungen mit hochgradiger Unstrukturiertheit, Wahrnehmungsüberlappungen sowie dynamischen Variationen (Beleuchtungsverhältnisse, Szenendynamik) zur modellieren.

TABLE OF CONTENTS

List of Figures

List of Tables

NOMENCLATURE

X	Location/place random variable
Z	Observation random variable
x_i	Element in sample space of X
z_j	Element in sample space of Z
m	Feature or observation distribution size; Index used with metric features/landmarks
H	Entropy (value)
I	Transmission/Transinformation (value)
N	Set of topological places (nodes); Place random variable
n	Number of discrete topological places (nodes)
N_i, p^t	Topological place (node)
S_i	Set of features extracted per node
F	Entire feature space; Feature random variable
$\mathbf{M}^t$	Topological model/feature-map
C	Set of ordered pairs indicating node spatial interconnection
f_i^*	Minimum relevant feature set per place (based on information-theoretic processing)
f^*	Minimum relevant total feature set/Relevant map
d	Cosine distance
f_j	Feature Category (sample of the discrete distribution of F)
O	Feature Cluster (codeword candidate) random variable
o_k	Element in sample space of O
Ω	Number of Feature Clusters
ψ	Number of Feature Categories
P	Precision (value)
R	Recall (value)
E	E-measure score function
β	Parameter of E
i	General index used with location/place
k	Parameter of k-means(number of clusters); General index used with Feature Clusters
j	General index used with observation and feature category
$\overline{S}$	Average Silhouette coefficient
$\mathbf{M}^h$	Hybrid topological-metric model/feature-map
{C}	Camera reference frame
{W}	World/object reference frame
{V}	Robot reference frame
X', Y', Z'	Camera frame axes
X, Y, Z	World/Object frame axes

ω, ϕ, κ	Roll, pitch, yaw angles of camera
${}^{\mathbf{W}}_{\mathbf{C}}\mathbf{R}$	Rotation matrix from {C} to {W}
$\mathbf{q_{off}}$	Offset vector between {C} and {V}
$\mathbf{q_r}$	3DOF Robot's position vector in 2D space
X_r	Robot *x*-position in {W}
Y_r	Robot *y*-position in {W}
θ_r	Robot orientation in {W}
α_m	Bearing/measured angle from robot heading to feature 'm'
$\mathbf{p_A^{im}}$	Position vector of landmark 'm' in node 'i'
X_A	Landmark *x*-position in {W}
Y_A	Landmark *y*-position in {W}
Z_A	Landmark *z*-position in {W}
M	Total number of metric features per node
(u_a,v_a)	2D image location of a given landmark *A*
(u_o,v_o)	Principal point/focal point of camera
$\mathbf{p_a}$	Projection vector of landmark A on image plane
$\mathbf{p_c}$	3DOF Camera's position vector in 3D space and Origin of {C}
$\mathbf{q_c}$	3DOF Camera's position vector in 2D space
X_c	Camera *x*-position in {W}
Y_c	Camera *y*-position in {W}
Z_c	Camera *z*-position in {W}
f	Camera focal length
r	Residual function
R	Residual functional
$\Delta\mathbf{q_r}$	Change in robot position vector
$\mathbf{J}$	Jacobean matrix/change of R w.r.t. q_r
J_{ij}	Element in Jacobean matrix/change of R w.r.t. q_r ($\partial R^i / \partial q_r^j$)
$\mathbf{v}$	Image point vector $(u_a,v_a,-f)$
d_m	Dynamics detection criterion
d_q	Quality-based metric feature selection criterion
f_g	Geometric descriptor in $\mathbf{M}^h$
f_g'	Geometric descriptor retrieved from $\mathbf{M}^h$
f_{ng}	Topological descriptor in $\mathbf{M}^h$
f_{ng}'	Topological descriptor extracted from view seen by robot

ACRONYMS

AGV	Aerial Guided Vehicle
BoW	Bag of Words
CCD	Charge-Coupled Device
COLD	COsy Localization Database
DFT	Discrete Fourier Transform
DOF	Degrees Of Freedom
DoG	Difference of Gaussian
EKF	Extended Kalman Filter
GLOH	Gradient Location and Orientation Histogram
GPS	Global Positioning System
GVG	Generalized Voronoi Graph
HEID	HEIdelberg Dataset
HMM	Hidden Markov Model
ICA	Independent Component Analysis
IFR	International Federation of Robotics
KF	Kalman Filter
KLT	Karhunen-Loeve Transform
LoG	Laplace of Gaussian
MCL	Monte Carlo Localization
MHT	Multi-Hypothesis Tracking
NN	Neural Network
PCA	Principal Component Analysis
POMDP	Partially Observed Markov Decision Process
RANSAC	RANdom SAmple Consensus
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localization And Mapping
SURF	Speeded Up Robust Feature
UAV	Unguided Aerial vehicle

To my family, with love and appreciation

To the determined generation of January revolution
& the new horizons for a newly born Egypt

Sherine

Chapter 1

Introduction

This chapter provides an overview on the scope and objectives of the thesis. It starts with a brief statistic about current autonomous robots. Autonomy concepts are introduced next, from which the main scope of the work is described. Afterwards, aims and objectives of the research are derived, with highlights on the main contributions that try to solve some of the current challenges and limitations in the state of the art.

1.1 Autonomous Mobile Robots; Recent Statistics and Today's Requirements

Autonomous mobile robots receive increasing attention on both levels of scientific and industrial or household appliances development communities. Several potential robot applications exist, ranging from dangerous tasks such as nuclear plants operations, de-mining and fire-fighting, to routine tasks such as industrial manufacturing, delivery of supplies in hospitals and airports, cleaning of offices and houses, and up to scientific missions like exploration of deep waters, planets and space.

According to the IEEE Spectrum [URL:SPECTRUM], there were already 8.6 million working robots at the end of 2008. This number certainly exceeded 11 million units in 2010

according to statistics of the International Federation of Robotics (IFR). Besides the leading multipurpose industrial manipulator robots within these numbers, service robots are currently standing with the increasing mass market. From a total of 63,000 units sold up to the end of 2008, the following rates have been registered for different *professional use service robots* [URL:IFR]: defense, rescue and security applications (31%), field robots (23%), cleaning robots (9%), medical robots (8%), underwater systems (8%), construction and demolition robots (7%), mobile robot platforms for general use (6%), logistic systems (5%), and with minor installation numbers counted for inspection systems and public relation robots.

On an equivalent parallel route, mobile robots are evolving extremely rapidly as human companions. They work their way into our homes in an attempt to fulfill our needs for household servants, pets and other cognitive robot companions. From these *personal and private service robots*, 4.4 million units were sold for domestic use (e.g. vacuum cleaning, lawn-mowing and window cleaning robots) and 2.8 million units for entertainment and leisure (e.g. toy robots, education and training robots). Within this category, the market of handicap assistance robots is still considered relatively small, but expected to double in the next four years according to the IFR. Robots for personal transportation and home security/surveillance will also be of high importance in the near future.

Mobile robots are becoming more involved in many fields for carrying out a variety of service-oriented tasks. Along with such extensive expansion, special attention needs to be paid to the design of these machines. The most important aspects and requirements expected to be met are skill and intelligence. The quest for developing *efficient* and *intelligent* robots has been and is still part of intensive research in diverse disciplines that principally include engineering, artificial intelligence, machine learning and machine vision. Figure 1.1 summarizes the latest IFR statistics for the recent and expected projected sales of service robots introduced above, while figure 1.2 shows some application examples for today's robots[1].

1.2 Concepts of Autonomy in Mobile Robots

An essential requirement for a mobile robot is the capability of autonomous navigation. Autonomous navigation is the process of safely moving a vehicle without human intervention

[1] Photos sources: (a) http://en.wikipedia.org/wiki/Da_Vinci_Surgical_System (b) www.hocoma.com (c) http://www.iat.uni-bremen.de/sixcms/detail.php?id=1268 (d) http://marsrover.nasa.gov/home/ (e) http://archive.darpa.mil/grandchallenge/ (f) http://www.techscoop.com.au/world-cups-robots/ (g) http://www.fanucrobotics.com (h) http://world.honda.com (i) http://www.care-o-bot.de (j) http://www.foxnews.com/story/0,2933,319473,00.html (k) http://www.irobot.com (l) http://robotionary.com/robotics.

from a start position to a goal position along a feasible trajectory. This includes the automated tasks of *finding a route, guiding the vehicle along the route, updating the vehicle's position from time to time,* and *naturally avoiding collision with obstacles during motion*.

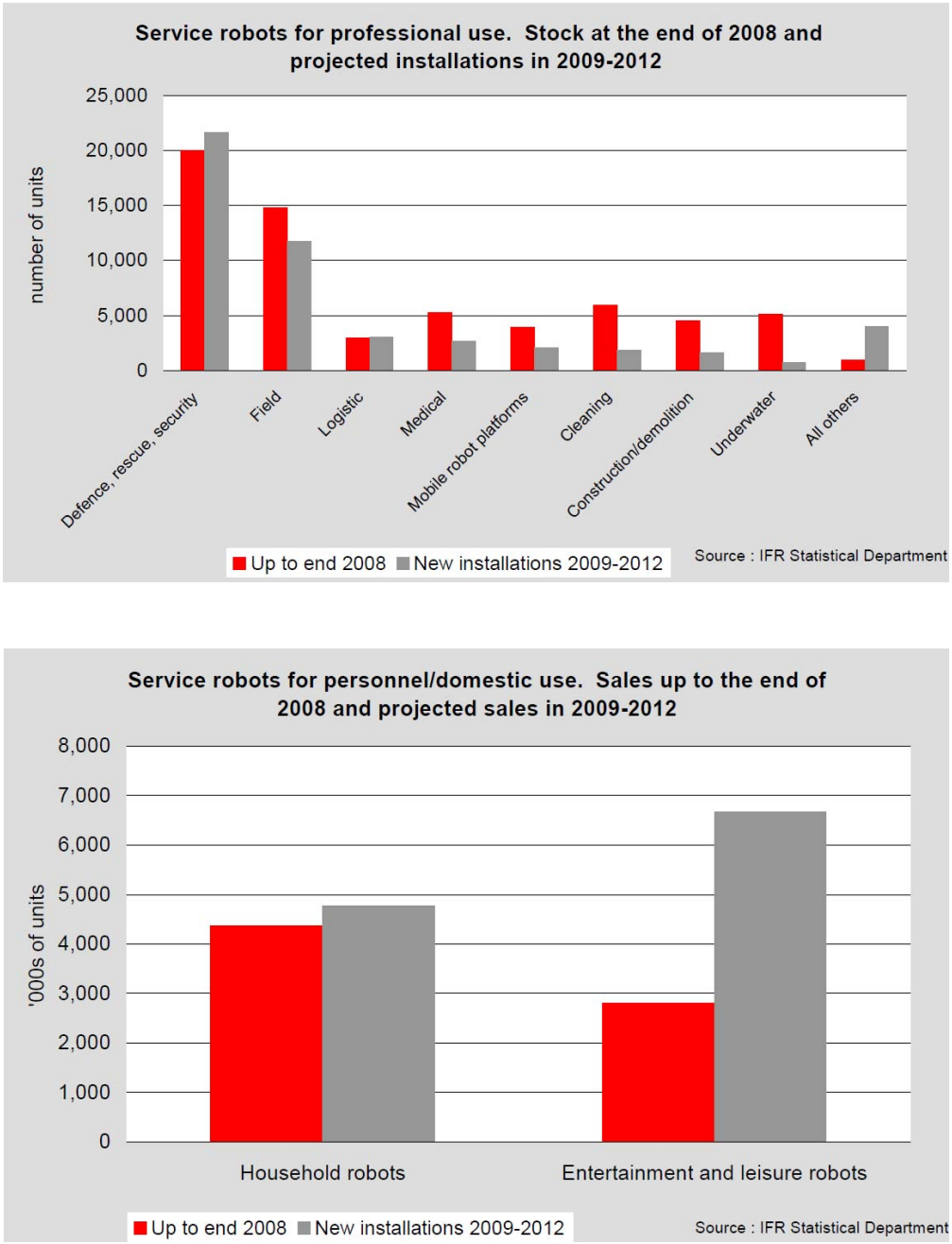
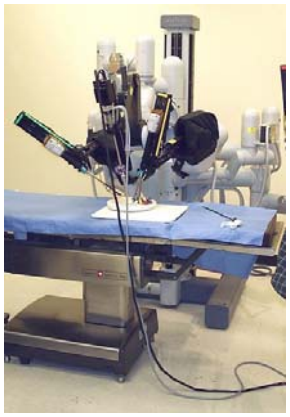


Figure 1.1. Service robots statistics for professional and personal use (Reference: International Federation of Robotics [URL:IFR]).



(a) Da-vinci surgery robot

(b) Therapeutic aid robots

(c) FRIEND III wheelchair rehabilitation robot

(d) NASA exploration rover on Mars

(e) Tartan obeying traffic regulations UAV, 1st place racer in DARPA grand challenge 2007

(f) Robot OFRO in surveillance mission at world cup Olympic stadium in Berlin, Germany

(g) FANUC M420iA robot in manufacture

(h) Asimo humanoid robot developed by Honda

(i) Museum robots for entertainment and tour guidance, Museum für Kommunikation in Berlin, Germany

(j) Yuki-Taro autonomous snowplow

(k) Roomba vacuum cleaner developed by iRobot

(l) Rescue robot participating in recovering efforts during the earthquake in Kashiwazaki City, Niigata

Figure 1.2. Application examples of today's robots.

According to [Leonard and Durrant-Whyte, 1991], the robot navigation problem is summarized in the three characteristic questions of: *(1) "Where am I?", (2)"Where am I going?"* and *(3)"How do I get there?"*. These questions are posed in the context of a description for the robot's world, or in other terms a general environment map. While the answer to the second navigation question appears trivial, since it is assumed to be given, answering the first and third questions represents a real challenge for autonomous mobile robots. Those questions, which are taken for granted by humans, are immensely complex implementations in robotic systems. Answering them binds the general navigation problem to several fundamental robotic tasks or behaviors[2], which are:

- Collision avoidance: mostly a reflexive behavior for avoiding collision with unexpected static or dynamic obstacles in the working space, without prior information about their shape or location.
- Local navigation: finding a way to the target in the absence of a detailed internal model about the environment.
- Self-localization: the estimation of robot's position with respect to its environment.
- Map building: the acquisition of internal models that describe the environment from sensor data.
- Path planning: finding a path which is interpreted as a sequence of motion actions in order to reach a goal from a current position. Path planning is substantially related to map building and self-localization. The current position, goal position and the complete map of free-ways are necessarily required to plan the trajectory towards the goal.

The previous modular tasks are listed according to a Recursive Nested Behavior Control (RNBC) structure proposed in [Badreddin, 1991]. This control structure provides an increasing bottom-up components integration for designing highly complex systems. Figure 1.3 illustrates this generalized design structure with pre-defined interfaces between the different behavior levels. The generality of the structure allows each level to be implemented using a sparse model on the suitable level of abstraction. The recursiveness enables ease of communication between the levels and makes it bounded. The nestedness embeds the lower level behaviors into the higher ones, in order to ensure stability and predictability of the

[2] Tasks are executed sequentially, while behaviors simultaneously.

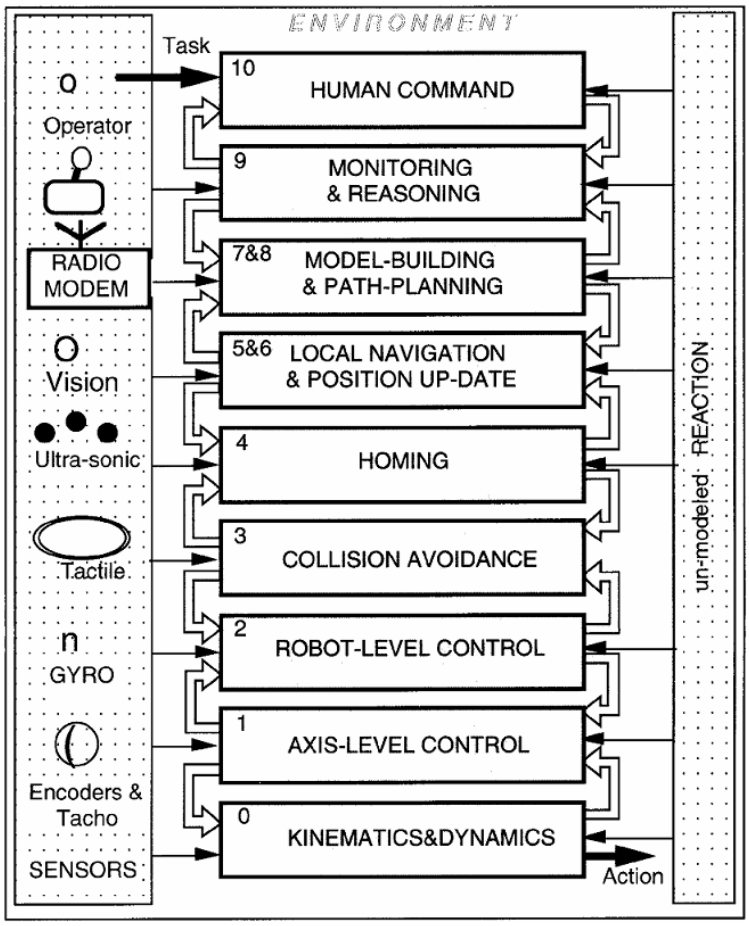


Figure 1.3. RNBC control structure [Badreddin, 1991].

system, and such that relatively fast behaviors are set at lower levels, with the more sophisticated and intellectual behaviors at higher levels. Thanks to those properties, a flexible and highly dependable design structure is afforded.

Despite the apparent generality of the RNBC structure, successful robot architectures suggest the use of simple behaviors at the lower operation levels to provide fast response suitable for dynamic environments (e.g. memoryless reactive behavior collision avoidance). Higher level behaviors like map building and task planning should be central and should rely heavily on internal environment models with a certain complexity that is proportional to both the task(s) to be achieved and the environment itself. Equipped with such a model, the robot can consequently plan movement, manipulation of the environment, or whatever action the task requires.

The work of this thesis is primarily concerned with environment map building for navigation purposes. It attempts to find suitable answers to the characteristic navigation questions *"What does the world look like?"* and *"Where am I?"*, which correspond to the 6th and 7th high behavior levels in the RNBC structure. Answering questions about map building and localization is challenging for a mobile robot. Several difficulties arise, such as dynamics of the environment, inaccuracies in perception, inaccuracies in localization, stationary and moving obstacles that are not considered in the map, resolution and accuracy of the sensors employed, etc. Being higher levels of certain sophistication in a complex and hierarchical structure example like the RNBC, an additional challenge counts, which is the efficient implementation of those navigational aid tasks or behaviors with the least possible complexity.

1.3 Motivation

For several decades, the practical robotic applications remained limited to carefully controlled environments such as assembly lines [Hamner, 2009]. Automation of the vast majority of real-world tasks remained out of reach, because our best systems could not handle the complexity of unconstrained environments. Nowadays, autonomous robots move beyond factory arenas and evolve rapidly as professional service providers and companions everywhere (e.g. museum tour guide, rescue, servant, and wheelchair robots) [Prassler, 2001; Gross et al., 2009]. They step into exploring and navigating inside new spaces which are full of unknowns, dynamics and unstructuredness. Several challenges confront those machines and their application fields, in which robots are required to accomplish their missions safely and efficiently.

For the safety and efficiency of navigation and manipulation, the future of robots, especially as service providers, is highly dependent on their abilities to understand, interpret and represent their working environments in an efficient and consistent fashion, and preferably in a way compatible to humans [Vasudevan et al., 2007]. Humans and even most animals including insects can handle their environments with ease, a task that robots are unfortunately far from doing with the same efficiency up till now. For example, a human can recognize an office environment of two different persons easily, though they might appear almost the same. Similarly, a moving few things around in the environment only makes a human notice that things have changed, but does not make one believe that he/she is in a different place. In the same way, one is able to recognize a place which he/she might have

previously visited at some other time or season even though the scenes are fundamentally different. These are issues that a robot cannot presently carry out with the same effectiveness as humans. It is not a question of computation power, as resources nowadays are adequate. It is a matter of the inherent recognition capabilities which are granted to humans and are limited in machines, and which research communities have been imitating since many years in a trial to minimize the human-machine intelligence gap. These scene complexities (e.g. similar places, distinctive landmarks or features apprehension, and continuous dynamics such as different objects that change their positions, moving people or vehicles that obscure the horizon, light that changes between darkness and full brightness, or nature that changes in appearance with the season), remain a consistent challenge for autonomous robots when they confront building a model for an indoor or outdoor environment and localizing themselves in it. They have to deal with such complexities in some robust means.

Figure 1.4 pictures a critical technical vision problem known as *perceptual aliasing*. It describes the case that different classes (e.g. places, objects) may have similar sensory images or induce similar descriptive patterns. Such perception problem is not specific to images only but for any sensory pattern. The two images in the figure show an elevator view at two different floors that one can almost say they look exactly the same. Perceptual aliasing is associated with another reverse problem. This is the *data association* or *correspondence*. Since a single pattern may match to more than one class, it is extremely difficult to determine the identity of a given pattern from those patterns exhibiting similarities. Figure 1.5 depicts a third problem known as *image variability*. It refers to the case that the same place may have different sensory patterns on different occasions, for e.g. when influenced by illumination

Figure 1.4. Different places appear the same – perceptual aliasing problem. Example images are taken at Building A-Mannheim University.

Figure 1.5. The same place appears differently – image variability problem. Example images are from the KTH-IDOL2 Database [Luo et al., 2006].

conditions or other dynamics introduced to the environment. The figure shows a corridor at three different times (sunny, night and cloudy conditions), and with a person interrupting the scene. The place is said to undergo general dynamic variations. Although the place is exactly the same in the three images, it can be falsely recognized by current recognition techniques. The two problems of perceptual aliasing and environment variability are complementary and stand in the way of obtaining robust recognition and reliable place identification.

The previous discussion summarizes a relevant perspective about environment modeling and localization challenges. Uncertainties exist and at any time (in both measurements and the environment itself as outlined by its scene complexities), and no or only partial knowledge about the environment might be available for the robot to provide some aid.

A second challenge perspective is outlined, which concerns the rapidly flourishing technological revolution. Several sensors with different modalities, as well as numerous feature extraction techniques and algorithms, are afforded nowadays, all of which permit possibilities for environment perception and modeling. The high resolution provided by a sensor (e.g. vision) reveals more complexity about the environment, which is not an easy issue to handle when considering the building of a compact model. Massive data are provided by sensors and feature extraction, but unfortunately accompanied with redundancies and non-necessities. Excessive features do not contribute to meaningful information, but actually degenerate the model's quality. On the one hand, irrelevant data will simply increase uncertainty and affect the accuracy of the task employing them. On the other hand, the size of employed features will increase the computational complexity of task, since the processing spaces (memory and CPU) are related to the environment model size. The higher the dimensionality of employed features, the higher the computational cost involved.

Consequently, the quality and efficiency of environment modeling is not interpreted by the blind use of multiple sensors or highly sophisticated techniques for perception and data modeling (e.g. shape-based analysis). A rather high relevance should be laid on investigating the quality and quantity of information to be perceived and processed, and in accordance with the available sensors, resources, and the investigated task. This information should fit to the amount required by the task; evidently not higher. In such a way, it is more meaningful to express a robotic system by its storage and processing units' consumption, and not just in terms of their numbers and sizes.

From a third and final perspective, navigating robots nowadays are working in growing environment spaces. Most robot metric navigation methods are implemented for small to medium-scale environments. Applying those methods to large-scale environments requires processing storage and computations proportional to this large space, which is not practical. This motivates the need for new solutions that adapt to large environments. Such solutions have to be presented in a suitable organizational structure that stimulates accuracy and computational efficiency performances.

The introduced perspectives represent real challenges for mobile robot navigation. It is firmly believed that the extent to which robots can navigate efficiently in their operating environments is decided by the way they represent their environments and localize themselves through accurate and economic methods. Figure 1.6 summarizes the three challenge perspectives with the technical terms introduced above, and which are tackled by this thesis.

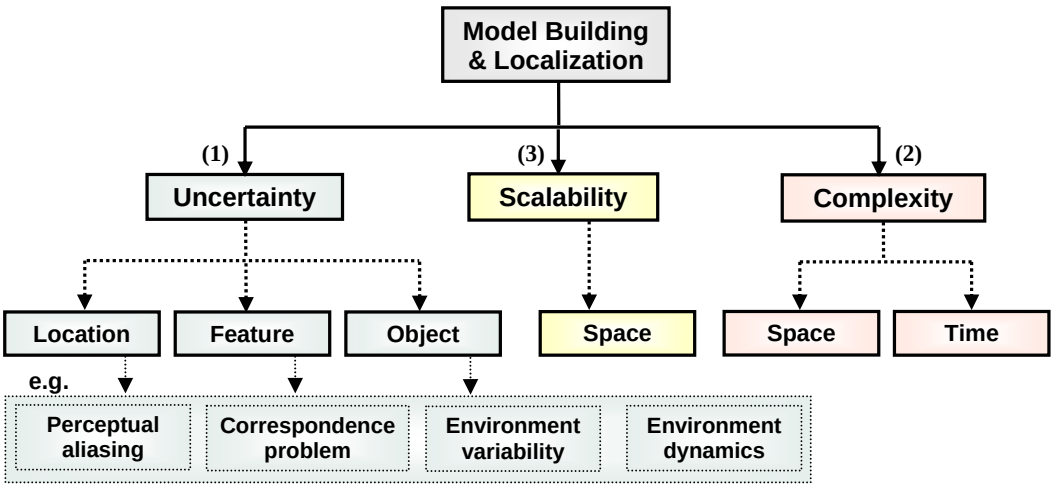


Figure 1.6. Challenges confronting environment model building and localization.

1.4 Objectives and Contributions

The major objective of the thesis at hand is to contribute to the development of *robust* and *efficient* environment modeling and localization for robotic applications. The efficiency in modeling concerns acquiring the most informative data and their representation in simplest form, whereas in localization concerns enhancing accuracy and computational efficiency performances. The robot's operating environments may be characterized by having clutter, unstructuredness and moderate dynamics, in addition to being large. In the frame of the defined objective and operating conditions, some related problems arise which are posed in the form of the following questions:

(1) How does the robot choose data that constitute importance, build the appropriate model for the environment and globally localize itself in it with a maximum bounded performance[3]?

(2) How does the robot maintain robustness of the environment model and the localization in a moderately dynamic environment?

(3) How are the model and the localization adapted for large space application?

The first question is a major subject in this work. To answer it, an *information-theoretic modeling approach* for the environment is proposed. The approach regards the environment as a collection of properties (i.e. features), each of which delivers a certain transmission value that contributes to the minimization of location uncertainty. The higher this transmission value, the more relevant the property is. Filtering properties with the highest transmission values and discarding those with the lowest transmission values, it is possible to generate an environment model that asserts high global localization accuracy and computational efficiency performances. Integrating data compression techniques in the solution approach is proposed to provide additional enhancement for the computational efficiency performance.

The merit of employing an information-theoretic modeling for the environment is that the modeling is independent of the employed sensor type and feature extraction methodology. It is only through statistical analysis that decisive conclusions can be generated about the data

[3]Performance has upper bounds which are a function of the environment data and in other terms the employed feature extraction methodology. Incorporated environment data itself can have bounds, as being a function of the specifications of the available resources (e.g. sensor resolution, memory unit size, processor power).

selection that achieves a double effect of minimizing uncertainties and reducing the data size. This issue enhances the general task execution by fitting the exact amount of information to the localization task, and with minimal data representation. Moreover, the approach can adapt a maximum bounded performance in accordance with limited available resources.

The second and third questions are answered together in the proposed information-theoretic modeling approach by adopting a special solution structure. This structure asserts robustness and scalability of the modeling and localization solution. The structure establishes a 2-level hierarchy, in which hybrid features are preserved. These hybrid features can be viewed in two different resolutions (coarse and fine) at the two levels, in order to support topological and metric localization modalities. Features are robustly detected and recognized under different conditions (e.g. viewpoint, scale, transformations and distortions). Dynamic variations, such as scene dynamics and illumination conditions, do not significantly influence the accuracy of localization.

The first level of the hierarchy deals with a granular data representation and confines localization to a coarse topological place (i.e. topological localization). Special requirements have been imposed for the environment model at this layer. It is valuable to have minimal number of relevant features for less computations but precise decisions (i.e. a compromise of accurate prediction and space and time complexities). It is also valuable to include robust features which are relatively insensitive to noise, dynamics and other errors. Such requisites urge the application of the information-theoretic approach at this level. Local feature extraction is employed in simple supervised information-theoretic-based learning to evaluate extracted features. The evaluation searches for those particular features that provide biased discrimination between topological places, in order to efficiently differentiate between them. Those features are selected and compressed concurrently to generate an information-rich and compact topological model. Such a model, which conforms to a high-level representation, can assist fast, accurate and robust robot topological localization and navigation. With the high attained topological accuracy, a second level of hierarchy is reliably extended from this level, in which it guides a second scalable metric localization, with a smaller projected space search.

The second level of the hierarchy deals with a finer data representation and localizes the mobile robot on a more precise level (i.e. geometric localization). Features at this level comprise the same topological data but in their non-compressed format to preserve identity.

Additionally, they are tagged with their metric position information, yielding a hybrid feature-map. Important requirements for localization at this level are: to accelerate localization and to minimize localization error. The first requirement complies with the proposed hierarchical structure design which provides speed and scalability. To fulfill the second requirement, two criteria are presented within a triangulation method for metric localization; one criterion to detect environment dynamics and feature mismatches, and a second to choose between features that contribute to less localization error. Consistency of the hybrid map is preserved at this level through the detection of dynamic features. Monitoring them on the running system for long terms provides decisions to exclude them from this level, or else they provide wrong estimations.

The proposed solution and methods offer enhancements for robotic computational efficiency without degenerating the accuracy. They satisfy performance demands, which are crucial for the efficiency of navigation systems, especially those aiming to work in large spaces and real-time. In particular, the following contributions are presented by this thesis:

- **An information-theoretic-based environment modeling approach for localization.** The approach evaluates the environment features and filters the most relevant (i.e. *highly discriminative*) ones based on their measured transmission property. Highly-transmissive features contribute to minimum localization uncertainty. The modeling and localization approaches are: (1) *robust* against location ambiguities, perceptual aliasing, illumination conditions and scene dynamics, (2) *efficient* in terms of online computational complexity, (3) *accurate* for localization as much as the feature extraction methodology provides, (4) *generic* as the modeling does not adhere to a specific sensor or an environment characterization. The proposed modeling has been evaluated and validated in two different indoor environments and with different sensor configurations (perspective and omni-directional vision). It proves suitability for application in unknown, densely cluttered, and unstructured environments using the environment's natural features.
- **A codebook compression module** that compresses the model features more than ten times providing a compact topological model, which at the same time maintains the accuracy of localization. The module is easy to construct and flexible in its parameters.
- **A top-down hierarchical structure for combined topological-metric localization using a single hybrid map.** The proposed structure supports dual global localization

modalities; coarse topological and fine geometric, and scales to large environments with lower complexity. The localization scheme is independent of the robot motion model (i.e. dead-reckoning). A hybrid map is proposed in the structure based on the same feature set. The feature set has two different resolutions, where both geometric and non-geometric properties of the environment are used differently to resolve the robot location at the two levels of the hierarchy.

- **Environment dynamics detection and geometry-based landmark selection criteria** for increasing the accuracy and robustness of metric triangulation-based localization.
- **Detailed quantified performance analysis** for measuring the performance gain in terms of memory, localization time and localization accuracy, a matter generally neglected by researchers.

The contributing solutions and methods are summarized in figure 1.7. The figure shows how the proposed solutions and methods contribute to higher localization accuracy, lower complexity and increased robustness requirements, which form the principal foundation of efficient model building and localization. Topological model building and localization solutions are highlighted as bold blocks, whereas additional solutions adopted for hybrid model building and localization are highlighted as dotted blocks.

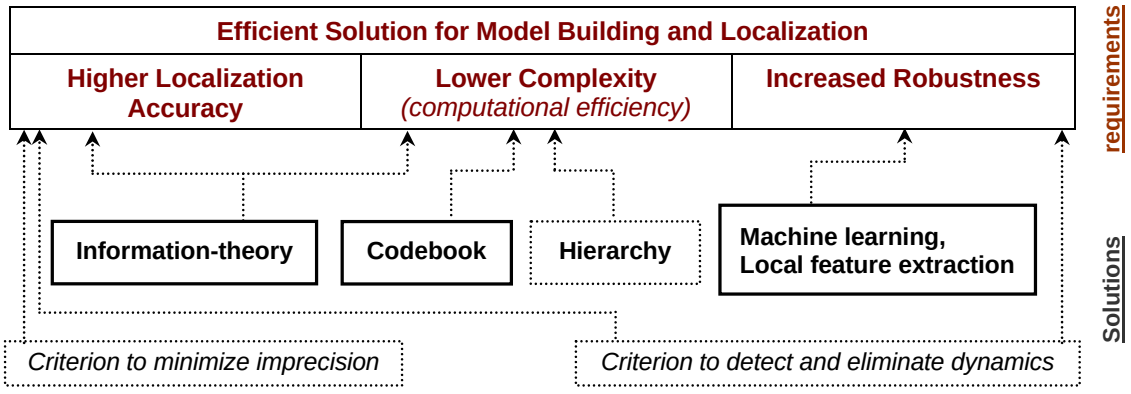


Figure 1.7. Efficient solution for model building and localization. An efficient model should provide robust and accurate localization performance with minimum computational complexity. The different contributing solutions satisfying those requirements are outlined. Blocks in bold satsify a topological solution, while additional dotted blocks satisfy an intregated topological-metric solution.

1.5 Thesis outline

The rest of the thesis is structured as follows:

Chapter 2 – Reviews research and developments related to the topic map building and robot localization. State-of-the-art is described, in which approaches are mainly classified into metric, topological and hybrid. The advantages and disadvantages of each approach are highlighted and some common drawbacks are outlined.

Chapter 3 – Presents basic preliminaries for the methods and techniques used. These include the topics: environment modeling and perception, sensors, vision-based modeling approaches, local feature extraction and evaluation, and finally the basic concepts and elements of information theory. The chapter ends with a concluding discussion on the motivations.

Chapter 4 – Presents essential contribution of this work, which is an information-theoretic appearance-based model for environment description. A general solution structure is outlined for model building and map generation, which is based principally on features evaluation and compression. The structure is presented within a topological context that forms a first layer in a hierarchical framework to be designed. The proposed information-theoretic solution makes use of an entropy evaluating criterion and a codebook concept, in order to produce a set of distinguished features called *entropy-based features*, with a second compressed format called *codewords*. The information-theoretic solution approach has been demonstrated using a vision sensor and local feature extraction. Thorough testing for employing both feature forms in a map to localize a mobile robot topologically is presented. Evaluating experiments indicate efficiency in localization and robustness against dynamic variations. High localization accuracy rates are obtained using the entropy-based feature map. Lower online complexity with maintained localization accuracies is realized using the codebook map. The attained performance gain and the substantial savings of the presented solutions are quantified.

Chapter 5 – Evaluates and validates the proposed information-theoretic modeling and codebook approaches on the COLD benchmarking database. The indoor benchmark database is constructed with an omnidirectional sensor and is subject to severe illumination and scene dynamics conditions.

Chapter 6 – Introduces a hierarchical framework that supports hybrid localization. The topological unit presented in chapter four is set as a first level of hierarchy and a second metric level is extended from the first. Precise metric pose estimation for the robot in 3-DOF is performed using a modified hybrid map. The map consists of a unified set of features

viewed in two resolutions; topological resolution in the form of a codebook and metric resolution in the form of entropy-based features additionally tagged with their position. Feature triangulation is applied to localize the robot given an identified topological place by the first level. Two methods are investigated for estimating the pose; The Newton-Gauss numerical method and the closed-form Geometric Triangulation. Additionally, two criteria are introduced in the triangulation to select features based on their stability and structure. Demonstrating experiments for the overall localization framework are given. They indicate accuracy and computational efficiency of the localization framework. Localization errors are bounded and accepted for an approach independent of dead-reckoning information. The proper choice of features based on geometry and exclusion of dynamic features show enhancement in the triangulation accuracy.

Chapter 7 – Presents final conclusions and summarizes the contributions developed in this thesis, highlighting their advantages and application fields, as well as their restrictions. An outlook to possible directions for future work is given.

Chapter 2

Map building and Localization: State-of-the-Art

Map building and self-localization are fundamental problems in mobile robots research. They set up two of the five preliminary tasks establishing the navigation problem, which includes additionally collision avoidance, local navigation and path planning. The topic has been studied intensively over the past years, to the extent that several solutions have been launched supporting different sensor configurations and a variety of application environments.

This chapter reviews the state-of-the-art of this topic. It summarizes the research efforts and the most successful approaches and underlying techniques that have been undertaken in the scientific area of robotic map building and localization.

2.1 Introduction

The modality of environment interaction in which the robot is engaged invokes a specific environment description that should facilitate this interaction. This need has launched several environment-mapping solutions to support both robot-environment interactions and robot navigation. The provided solutions span different application environments, and are founded

on different platforms and sensing configurations. Each solution reports pros and cons, with one solution being more suitable for a specific application and an environment than the others.

Map building is the process of acquiring a robot internal model about the environment through the robot sensors. The resulting model is termed the *map*. Building an appropriate map has been regarded as a major perquisite for the other navigational functions or behaviors; collision avoidance, self-localization and path planning. Some navigation approaches that do not require a map also exist (i.e. mapless navigation) [Desouza and Kak, 2002]. They depend on instantaneous relative measurements to find their ways and reach easily defined destinations. Those approaches are simple but cannot be applied if the robot needs to reach particular locations in a relatively complex environment. Therefore, a map is always regarded as an elementary requirement for mobile robots engaged in non-trivial tasks [Budenske, 1989].

Self-localization, or simply *localization*, is the ability of the mobile robot to determine its location in space. This location can be determined relatively with respect to a priori known location if the exact motion model of the vehicle is known. However, this is not the common case, since the motion model is accompanied by uncertainties. In addition, a robot's absolute location cannot be identified without a map. Knowledge of self-location and locations of other places of interest from the map is the basic foundation upon which high-level navigational operations, such as local navigation and path planning, are built. Therefore, obtaining a good performance from a navigation algorithm is strongly bound to accurate robot localization in the environment [Bonin-Font et al., 2008]. It is worth mentioning that without the notion of a location, the robot will be limited to a reactive behavior based on local stimuli only. This poses limitations and restrictions on the action planning capabilities of the robot.

Human-constructed maps and external infrastructures, such as the Global Positioning System (GPS) have been proposed as navigation solutions. Those solutions are sufficient, but have their limitations. Manual feeding of systems that are supposed to be autonomous is a disadvantage, especially for a dynamic environment that will require map updates. The GPS infrastructure also might not always be available (e.g. for planetary exploration robots, many terrestrial environments such as indoors, near tall buildings, under foliage, underground and underwater). Additionally, the navigation problem, in many cases, has to be necessarily solved using the robot's internal sensors alone due to considerations of flexibility, costs, or other

requirements. Therefore, other solution approaches that mainly depend on the robot's attached sensors are adopted.

Sensors are the information sources that basically assist map building and navigation solutions. Like humans and animals, the robot uses its sensors to perceive the surroundings and get information about the environment through every observation and action it makes. A variety of sensors exists, which enables measuring and encoding various types of data modalities and information. Some sensors provide position information straightaway. That is to say, a direct solution for the localization problem is obtained. Other sensors provide other measurement modalities from which the position information can be inferred. In general, two distinct robotic sources of information can be distinguished [Filliat and Meyer, 2003]:

Idiothetic (Proprioceptive) sources – They provide internal information about the robot movements. Those movements comprise speed, acceleration, orientation, leg movement or wheel rotation. Typical sensors include gyroscopes, accelerometers, compasses and wheel encoders. Integrating idiothetic information over time results in a position estimate known by *dead reckoning, path integration* or *odometry*.

Allothetic (Exteroceptive) sources – They provide information about the environment and are robot external sensors. Examples are vision, microphones, odor, tactile, sonars, laser range-finders and scanners. Allothetic sources play the main role in perceiving the environment for navigation and correcting the robot internal states.

The advantages and drawbacks of the two information sources are complementary. The major problem with idiothetic information lies in the nature of the integration process. It is subject to *cumulative errors* that can grow quickly. Causes of the errors are systematic because of the mechanical and electrical construction (e.g. unequal wheel diameters, misalignment of wheels, finite encoder resolution and finite sampling rate). Additional non-systematic errors occur in the joint and the task space (e.g. floor) due to the effects of wheel slippage, uneven ground, variation in the contact point of the wheel, and probable collisions. Both sources of errors lead to a continuous decrease in the quality of movement detection [Borenstein et al., 1997; Siegwart and Nourbakhsh, 2004].

Figure 2.1 shows an example that manifests the position errors obtained from dead-reckoning. It is demonstrated that such information is not reliable over long periods of time.

Shown in the figure is the robot's path as obtained by its odometry relative to the real environment map. Over long periods of time, turn and drift errors[1] as well as translational errors accumulate continually. Orientation errors dominate because they grow without bound into translational position errors, the issue that causes misalignment of the odometry measurements relative to the real map. The additional use of a heading sensor (e.g. gyroscope) can help to reduce those severe cumulative errors, though not totally avoiding them.

In contrast to idiothetic sources, the quality of allothetic information is stationary over time [Filliat and Meyer, 2003]. Nevertheless, this information suffers from the problem of *perceptual aliasing*. It is the case that, for a given sensory system, two different environment places can induce similar perceptual patterns. Therefore, a good strategy is to fuse both kinds of information sources to overcome problems from both sides. This will be encountered in the literature survey. In such a case, allothetic information helps in the compensation of idiothetic drift, and in turn idiothetic information supports the disambiguation in allothetic place.

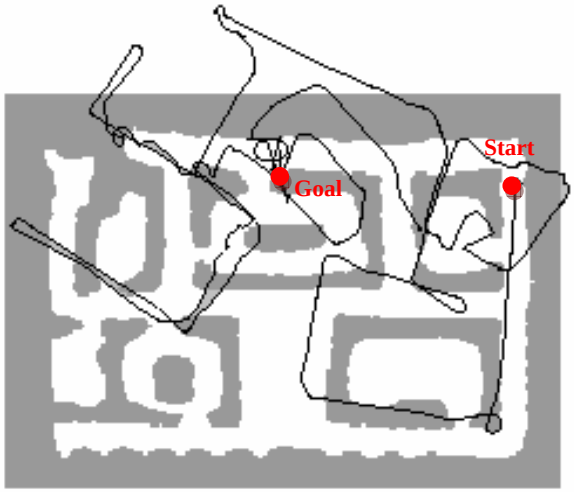


Figure 2.1: Example demonstrating position errors caused by dead-reckoning [Thrun, 2002].

The map building and localization have been often introduced and studied in the literature as a fused topic. The literature commonly classifies both into topological and metric approaches, in relation to the same adopted classification for space representation. The rest of

[1] Turn and drift errors are the angular orientation errors. Drift error is error in the orientation due to difference in the error of wheels [Siegwart and Nourbakhsh, 2004].

this chapter presents the related work for robotic map building and localization classified into metric, topological and hybrid approaches. The robotic map building is first surveyed in section 2, followed by the localization in section 3. Section 4 is dedicated to the recent hybrid map building and localization approach. The chapter ends with a brief summary for the state-of-the-art and highlights for some of the current limitations which motivate this work.

2.2 Map Building

Robotic map building addresses the problem of autonomously acquiring spatial representations of physical environments through mobile robots. This is interpreted into a process of locating certain entities (e.g. places, features, landmarks or obstacles) within a certain frame of reference. Accordingly, the constructed map is viewed as a list of any environment objects together with their descriptive locations [Thrun et al., 2005]. The robot uses mounted sensors to perceive the environment and build the map automatically. A map is constructed during or after an environment exploration phase. A second option is that the robot constructs the map during navigation, in which location of both environment data and robot are solved simultaneously. The latter is known as Simultaneous Localization and Mapping, or SLAM for short [Smith et al., 1990; Durrant-Whyte and Bailey, 2006].

2.2.1 Map Categories

Traditionally, maps are classified into two major distinctions: *metric maps* and *topological maps* [Meyer and Filliat, 2003; Siciliano and Khatib, 2008]. Nevertheless, other map categories exist, such as appearance-based maps, semantic maps and hybrid maps [Buschka, 2005].

Metric maps contain distance information that corresponds to the distance actually found in the real world. They are sometimes called geometric maps. Such map art contains, for instance, information about the length of a path, the dimensions of a building, or the metric position of an object. A scaled road map, a city map and a CAD model (see figure 2.5-a) are generic examples of metric maps.

Topological maps are representations in which the environment is modeled according to structure and connectivity. They can be represented by a graph, with nodes corresponding to key environment places and edges accounting for connections or relations between the places.

In such a map, one finds information about place connectivities and paths, such that one can decide which edges and nodes to pass to move from one place to another. A subway map is a typical example of a topological map, where each node represents a train station and each edge represents a train connection.

Appearance-based maps are maps in which sensor data are used to form a direct relation from the data to a specific position. One of such maps is described in [Kröse et al., 2001]. *Semantic maps* are used for making decisions at a high level. They contain information about environment objects, space properties and relationships [Galindo et al., 2005; Vasudevan et al., 2006]. *Hybrid maps* combine different map types (for e.g. a metric map and a topological map). A hybrid map has additional links that connect elements in one map to elements in the other map. If these links do not exist, the hybrid map degenerates into merely a collection of totally unrelated maps [Buschka, 2005].

Another historic taxonomy for maps is *world-centric* versus *robot-centric* [Thrun, 2002]. World-centric maps are represented in a global absolute coordinate space. Entities in the map do not carry information about the sensor measurements that lead to their discovery. In contrast, robot-centric maps are described in the measurement space. They contain sensor measurements a robot would perceive at the different locations. Robot-centric maps are easier to build, since no translation of robot measurements into world coordinates are required. Nevertheless, they suffer from a severe perceptual aliasing problem since obvious geometry is missing in the measurement space.

Table 2.1 summarizes the map taxonomies, as well as the map building problem. Metric versus topological is the conventional classification, and is by far the most common taxonomy in literature of both map building and localization. In an extended sense, the map and map building can be further classified according to the way the map is indexed; being location-based (i.e. occupancy) or feature-based [Thrun et al., 2005]. Aggregations on the same taxonomy level lead to a composite or a hybrid map. Generally, metric maps are constructed when high accuracy is required, for example in exact positioning and precise path planning. Application examples are Automated Guided Vehicles (AGV), which are restricted to fixed trajectories and paths, and are limited to repetitive tasks. On the other hand, topological maps are more adequate and much easier for autonomous navigation – especially indoors – which does not require such high precision.

Table 2.1. Map and map building taxonomies

Metric	*Versus*	*Topological*
World-centric	*Versus*	*Robot-centric*
Occupancy-based	*Versus*	*Feature-based*

Both map building and localization can be expressed against the spatial representation, in which the environment space can have two forms; continuous and discrete. Table 2.2 shows the metric-topological classification expressed against the continuous-discrete environment representation. Metric maps and metric locations are representable in both continuous and discrete forms of the space. Topological maps have only a discrete representation. Consequently, the map building requires the definition of a specific data structure, as well as a method to update the structure. This will be encountered in the next subsections which discuss building metric and topological maps.

Table 2.2. Map building and localization versus space representation

map building & localiazation \ *space*	**Continuous**	**Discrete**
Metric	√	√
Topological	×	√

2.2.2 Metric Maps

Metric maps describe the properties of the environment as either a collection of 'environment objects' or 'occupied positions' in space, with the geometric relationships among them preserved in a common global reference frame (normally the Cartesian space). Hence, two specific kinds of metric maps are identified; *feature maps* and *free space maps*. In the general sense, feature maps differ from free space maps in that the sensed data are first transformed into a higher abstract representation which is secondly geometrically indexed. Most of the existing metric map-building solutions are based on SLAM [Thrun, 2002]. The map building is not separated from the localization process, and consequently, difficulty arises because the

localization errors are incorporated into the map. Hence, those solutions rely on the robot kinematic model together with probabilistic filters to compensate the errors.

Feature maps contain landmarks or any environment features that the map depicts. Landmarks are references with known positions. They can be natural or artificial. Natural landmarks form high abstractions composed of simpler features. In structured indoor environments, it is often normal to encounter walls, corners and edges, which are formed by combining simpler geometric atoms (e.g. points and lines). Other higher abstractions exist like doors and windows. Distinguished environment objects also serve as natural landmarks (e.g. fire extinguisher, clock, wall painting, etc). Artificial landmarks or beacons are reference objects introduced to the environment, which mostly provide unique identification. They are often used when high positioning accuracy is required. Examples are bar codes and light-reflector tags. Other than landmarks, feature maps may employ less-abstracted features such as vertical lines or point features extracted from objects' surfaces, which are described by certain descriptor. Figure 2.2 shows a simple representation for a feature map for an indoor environment where walls and corners serve as natural landmarks.

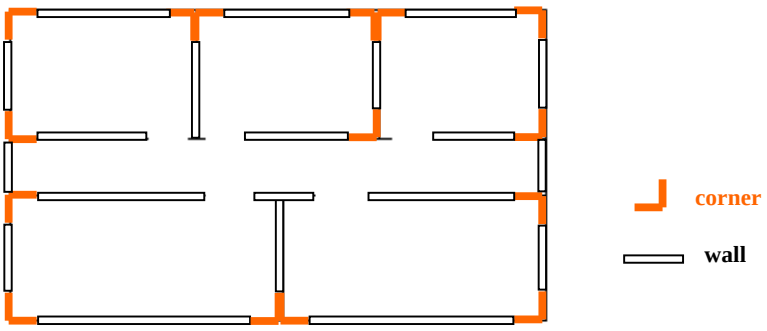


Figure 2.2: Simple representation for a feature-based map. Lines are used to represent walls and corners.

In feature-based map building, the robot has to estimate its unknown pose while calculating the position of the features simultaneously. Building the map this way requires that each feature is first extracted from the sensor data relative to the robot, matched against the features in the existing map, and finally the feature can be added if it does not exist, or updated if it exists, or used to correct the position estimate in the SLAM execution.

Feature-based maps have been constructed using line segments extracted from walls through sonar [Ohya et al., 1994; Crowley, 1989], and laser scanner [Borges and Aldon, 2004; Pfister et al., 2003], with the employment of Kalman filters. Additional features, such as corners and edges, were extracted using sonar in [Chong and Kleeman, 1999; Leonard and Durrant-Whyte, 1991]. Laser range finders have been used to map point features and register scans in [Biber and Staßer, 2003; Surmann et al., 2003]. In a similar manner, 3D point clouds have been extracted and registered for indoor and outdoor environments using laser range finders [Nüchter et al., 2006; Cole and Newman, 2006], stereo vision [Se et al., 2001], and combined laser-omnidirectional-vision sensors [Biber et al., 2005].

Free space maps are the second map type which represents the environment accessible parts only through tessellation. Tessellation can be uniform or non uniform. Figure 2.3 shows a planar workspace populated by polygonal obstacles. A map can be generated in the form of discrete non uniform cells obtained by exact cell decomposition. The method selects boundaries between discrete cells based on geometrical criticality [Siegwart and Nourbakhsh, 2004]. Decomposed cells can be rectangular, trapezoidal or convex polygon shaped. The figure shows trapezoidal decomposition obtained by extending vertical lines up and down in the workspace at obstacle vertices until they touch either an obstacle or the boundary of the workspace. The representation is compact because each cell is stored as a single node, resulting in a total of only eighteen nodes for the given example. Such exact decomposition is, however, not always feasible, because it is a function of environment obstacles and free space which are not necessarily sharply defined.

Another common alternative of free space maps are the *occupancy grid maps* [Elfes, 1990]. They present a fixed decomposition by discretizing the space into a regular grid, whereby each cell stores occupancy information about the area it covers. Each grid-cell is assigned a single value representing the probability that this cell is occupied by an obstacle. Grid maps are volumetric, in the sense that they offer a label for every possible location in the environment. Although features can still be indexed in a grid representation, it is more frequent that grids are indexed with occupancy. Figure 2.4 shows the occupancy map generated for the virtual workspace defined in figure 2.3, while figure 2.5 shows an actual metric map constructed using fine-tessellated occupancy grid and a laser ranger for the CAD environment on the left hand side.

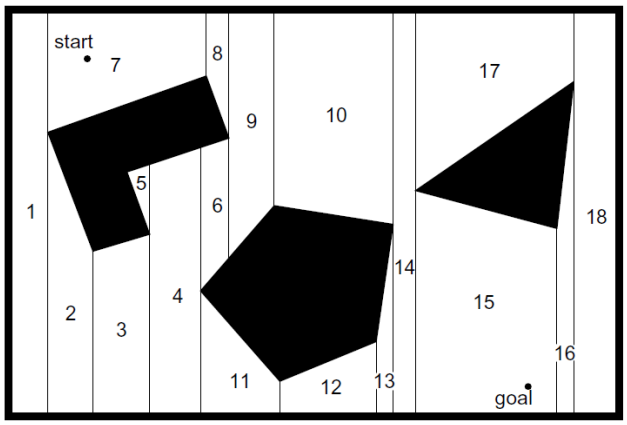


Figure 2.3: Example of exact cell decomposition mapping [Siegwart and Nourbakhsh, 2004].

Falling into SLAM categories, the occupancy grid map building proceeds by estimating the robot position first, and then the values of the grid cells are updated whenever the robot has a new set of data. A sensor model and an update rule are required. The sensor model describes the uncertainty of the sensor data, and the update rule describes the uncertainty of the resulting map. The robot's position during map building is estimated directly through the odometry values given by wheel encoders, or through the map which is built concurrently in the SLAM execution.

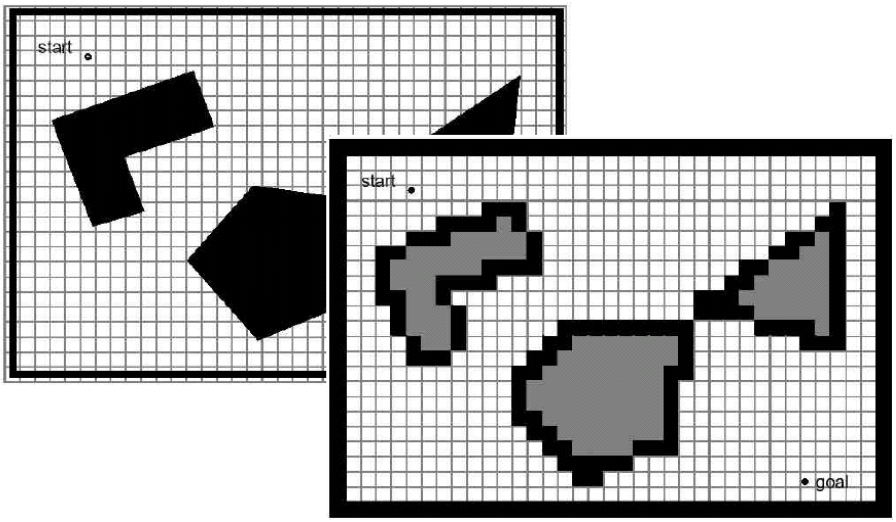
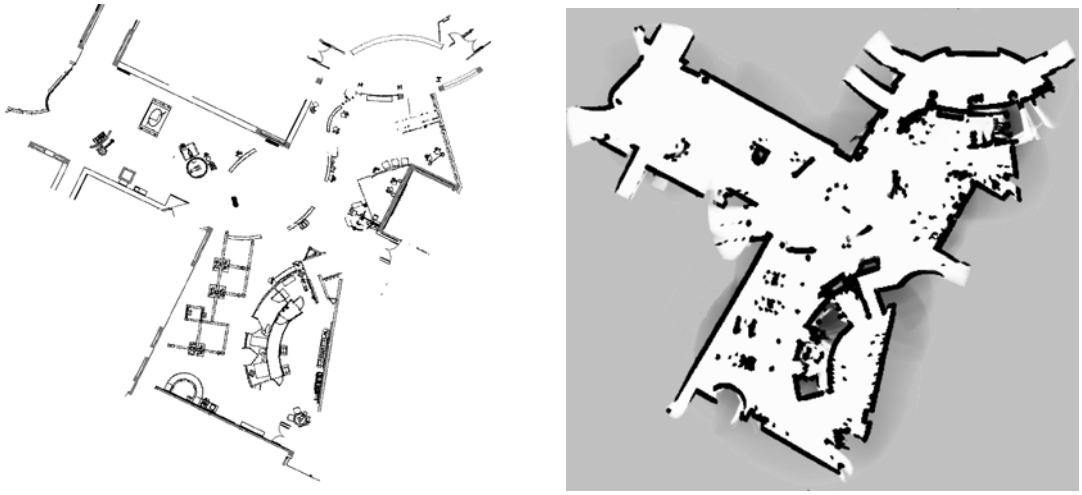


Figure 2.4: Fixed decomposition for the workspace shown in figure 2.3 [Siegwart and Nourbakhsh, 2004]. Cells in white are free space, black are occupied space, and gray are unexplored space.



(a) (b)

Figure 2.5: (a) CAD Map of a large open exhibit place (b) Occupancy map generated using laser range data. A grid cell probability of occupancy is defined by a value bounded by (0,1), where 0 indicates definitely not occupied (free space) and 1 means definitely occupied. A prior probability of 0.5 is always initialized, which implies unexplored space (depicted by the light gray regions) [Thrun et al., 2005].

Most grid map implementations are range sensors based [Thrun, 2002]. Early implementations used sonars [Elfes, 1989; Thrun, 2003], while more recent ones use laser range finders [Yguel et al., 2006]. 3D grid maps have also been constructed from disparity information using stereo vision [Chen and Xu, 2006], and a combination of vision and proximity [Bennewitz et al., 2006].

Occupancy grids have proven to be simple and quite useful for obstacle avoidance and small-scale planning purposes [Elfes, 1989]. They are suitable for coverage path planning (e.g. a field insecticide spraying Unmanned Aerial Vehicle (UAV) and a floor-cleaning robot). Performance-wise, they offer a constant time access to grid cells and can model unknown (unobserved) areas, which is an important feature in the context of exploration. However, these models become difficult to handle when the environment size is large. They require a lot of memory resources, and much of the detailed information stored in the cells becomes irrelevant and creates a corresponding data association problem.

Consequently, a significant difficulty arises with occupancy grids working in large environments as a tradeoff between the grid resolution (granularity) and the computational complexity. Ideally, the grid size should be as small as possible (fine grained) to capture environment details and facilitate accurate position estimation. From another side, a large grid

size (coarse grained) may be necessary for feasible computations, given that storage and computation increase in proportion to the number of grid cells. Thus, it is evidently concluded that fine grid resolutions are not suitable for large-scale environments. Tasks, like path planning, become computationally expensive with them. Some methods to obtain variable granularity have been proposed [Burgard et al., 1998; Nebot and Pagac, 1995], and hence they focus the resources at regions of environmental complexity. Nevertheless, the methods possess implementation difficulties of their own [Bailey, 2005].

Occupancy grids and feature maps have been contrasted to each other [Laaksonen, 2007; Bailey, 2005]. The advantage of feature maps over occupancy grids is that they contain only the needed features, and hence are much better to process. Furthermore, map update is easier. To correct a grid map, the entire map has to be recalculated, while in feature-based maps only the position of affected objects has to be adjusted. Occupancy grid, on the other hand, has the advantage that it shows immediately if a certain location on the map is accessible and free. For example, a position inside the wall is occupied in the map, so the robot cannot be in such a location. With feature maps, extra processing is needed to get some kind of information like that out of the map. In such reasoning context, feature maps do not facilitate path planning and obstacle avoidance, and the tasks must be performed as separate operations.

2.2.3 Topological Maps

Topological maps are abstracted maps that contain vital information only, with unnecessary detail removed. They form abstract but concise representation and are less concerned with the geometric properties. Generally, these maps lack scale. Distance and direction may vary but the relationship between points is always maintained.

Building topological maps was introduced into the field of robotics following studies of human cognitive mapping undertaken by [Kuipers, 1978]. Since then, much progress has been made in the field, especially using vision. Robotic topological maps are inspirations from biological behaviors which do not hold a detailed metric representation of space. Humans and animals memorize environments by recognizing landmarks and relating them to places, as well as relating places' adjacencies. Therefore, a robot topological map is a graph-like structure that models the environment as a collection of spatial nodes with interconnectivities.

Figure 2.6 shows a graph representation for a topological map where significant places in the environment form the graph *nodes* and place neighborhoods form the graph *edges*. Interconnections describing the map can interpret other meanings other than adjacency. They can indicate actions or behaviors, such as ‘traversing a door’ or ‘wall following’. Furthermore, they can indicate specific relative distances and angles. Topological maps are much easier for robot navigation than metric maps, since they are at a higher level of abstraction. Finding a path from one point to another on the map only requires finding a possible traversable path between two nodes following the arcs. If the interconnecting edges are annotated with additional metric information, path planning is a straight-forward task using a readily path planning algorithm, such as Dijkstra’s Algorithm.

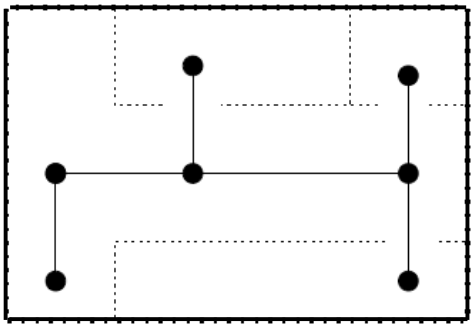


Figure 2.6: A graph representation for a possible topological map.

As mentioned, nodes are defined at some distinctive places. This highlights two issues for the topological node: what is a place and how can a node in practice be assigned.

The term ‘place’ has been defined in some literature. [Chatila and Laumond, 1985] defines the place as an area of a functional or topological unit. Examples of topological units are rooms and corridors, whereas a printer, for instance, is a functional unit. In this perspective, connectors used to connect the places are restricted to doors, stairways and elevators. A place is also defined as a nameable segment in a real world environment, which is distinguished due to different functionalities, appearances or artificial boundaries [Pronobis et al., 2010]. Furthermore, a place is a collection of physically close positions which share similar visual appearances [Werner, 2010]. The three definitions, in a global sense, imply that a place means a set of features that share a common representation, and which are close in space. This consideration is compatible with the human comprehension of places.

In practice, there exist two possibilities to assign the nodes [Valgren, 2007]. The first is to let a human operator assign them (i.e. by guiding the robot and telling it when a new node should be assigned by for example, pressing a button) [Thrun et al., 1998; Ulrich and Nourbakhsh, 2000]. Different supervised learning algorithms can be applied later, such that the robot eventually learns where to define the nodes itself [Radhakrishnan and Nourbakhsh, 1999; Mozos et al., 2006]. The second possibility is to let the robot assign the nodes automatically. The robot detects abrupt changes in the environment (or rather abrupt changes in the sensor input data stream) that signal a new node occurrence. Changes in the robot actions or behaviors help to signal the new node detection (e.g. turning a corner or entering a door). For instance, assume a robot is moving in a corridor and executing a free space or wall following behavior. It reaches the end of the corridor or the wall after a while and senses this by its range sensor. The robot will mark the place as a node, and will then execute an action or a behavior to continue moving. Some other ways monitor the disappearance of sensed data patterns or objects to detect the transition into a new place [Tapus and Siegwart, 2005], while others assign the nodes by clustering data [Kuipers and Beeson, 2002]. Both ways are mostly vision-based. Finally, a much simpler way other than all the previous ones is to assign the nodes after the robot has traveled either a distance far enough from the previous node or a fixed distance [Andreasson and Duckett, 2004; Goncalves et al., 2005].

The second possibility for node assignment is obviously more appealing for full autonomy. Despite that, it is common for indoor environments, and especially with vision sensors, to manually assume that each hallway or room is a node, and execute fixed distances for the node assignment. In outdoor environments, the natural segmentation of rooms and corridors does not exist and human classification represents the world in other topological terms, such as ‘in the parking place, ‘close to the fountain’ or ‘outside the restaurant’ [Valgren, 2007]. Supervised learning methods are promising solutions to identify the semantic regions the same way as defined by humans [Mozos et al., 2005; 2007]. This complies better with the correct definition of a place, which might be missed by the automated techniques.

Topological maps are constructed in various ways depending on the available sensors. Typically, they are feature-based maps, though not compulsory. It is more efficient to use sensor allothetic data to characterize the nodes with *fingerprints*, while use sensor idiothetic data to induce the connectivity relations between the nodes [Remolina and Kuipers, 2004].

In order to generate the whole topology automatically, navigation solutions can be employed. Those solutions are basically range sensors based, and define different structures of space such as waypoints, navigational areas or convex polygons. These structures are explained in the scope of the methods; visibility graphs, Voronoi diagrams and cell decomposition [Giesbrecht, 2004]. Since a major disadvantage of the idiothetic data is the perceptual aliasing, it is preferable to combine the generated nodes with additional allothetic data characterization. In the general view, generating topological maps this way undergoes two typical strategies [Shatkay and Kaelbling, 1997]; building the map directly from sensor data or deriving it from a metric map through some process of analysis.

Building the map directly from sensor data can be implemented using different methods. The most common method is the Generalized Voronoi Graph (GVG) [Siciliano and Khatib, 2008]. The GVG derives a route network from information about the obstacle boundaries, from which the topological nodes automatically emerge. The graph is formed by the set of points in the free space, which are equidistant to the n closest obstacles in the n-dimensional space. Figure 2.7-a shows a two-dimensional space environment and the corresponding GVG (fine lines) consisting of curves that intersect at meeting points and end up in the corners of the environment. GVGs have been extracted directly from range sensor data in [Choset and Nagatani, 2001; Wallgrün, 2004; Beeson et al., 2005; Thrun, 1998; Blanco, 2000].

Topological maps can be also automatically generated from visibility graphs [Siegwart and Nourbakhsh, 2004]. Those graphs consist of lines of sight which connect polygonal obstacles, without crossing their interior. Before applying such procedure, the environment obstacles are usually enlarged by a size equal to that of the robot (i.e. diameter or longest dimension). Topological nodes are next assigned to the midpoints of the visibility connections. In a similar manner, the exact cell decomposition discussed before (see figure 2.3) generates a topological map, in which the free space is represented via convex or trapezoidal polygons [Siegwart and Nourbakhsh, 2004]. Nodes are assigned either to the midpoint of the constructed lines or to the middle of the generated polygons.

Both the GVG and visibility graphs are known as roadmap methods used for path planning. That is why they form a geometric representation of routes rather than a graph representation. But they can still derive the topology automatically.

A third method to generate the topology directly from sensor data is to use the route generalization method [Lankenau and Rofer, 2002]. Figure 2.7-b shows the locomotion of the robot as recorded by its odometry in solid curved line. Odometry can be used to detect the environment topology. The rectangular boxes in the figure represent the so-called 'acceptance' or 'navigational areas'. As long as the robot remains inside such area, it is assumed that it is still located in the same region. Navigational areas generate the edges of the topology, while nodes are generated at the intersections of those areas. The method is suitable for mapping corridor-rich environments, where nodes emerge at hallways, corners and junctions; and edges represent the straight corridors.

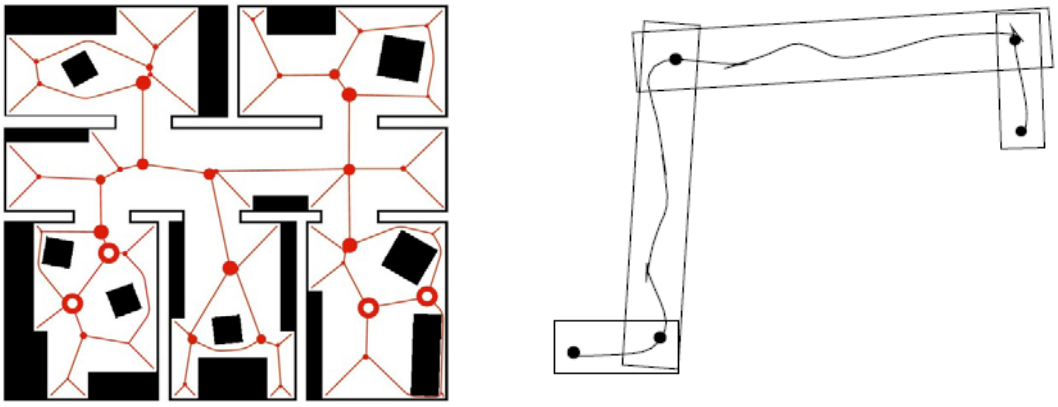


Figure 2.7: Left: Generalized Voronoi graph (GVG) [Wallgrün, 2004] with vertices placed at the position of the corresponding meetings points, based on equidistant edge generation from obstacles. Right: Route generalization [Lankenau and Rofer, 2002]. The figure shows the locomotion of the robot as recorded by its odometry, with the detected corners and the acceptance areas for each route segment.

The second strategy to build a topological map automatically using navigation solutions is to extract it from another type of map, usually a metric map. In [Thrun and Bücken, 1996], a grid map is preprocessed by setting the emptiness and occupancy values to crisp values through thresholding. A Voronoi diagram is next constructed on the empty space of the thresholded grid map. Local minima found on the diagram are used to partition the metric map into nodes of the topological map. Another procedure based on a thinning algorithm has been used in [Choi et al., 2002]. The procedure eliminates occupied cells gradually until the skeleton of the structure appears. Nodes are selected at the end points of an arc, at the corner points where an arc slope varies, or at the branch points where more than three arcs intersect.

The technique represents a line-intersection modeling of space where intersections denote the topological nodes. Clustering and graph cuts have also been applied in [Zivkovic et al., 2005; Valgren, 2007] to let the nodes evolve as clusters with pattern similarities. Other methods apply learning through a hidden Markov model (HMM) to fit the data [Shatkay and Kaelbling, 1997], or extend the HMM with robot action data into a partially observable Markov decision process (POMDP) [Koenig and Simmons, 1996; Shatkay and Kaelbling, 2002; Tapus and Siegwart, 2005].

Topological maps have several advantages. Since they are abstract maps, they offer a simple representation for the space that can be easily used in high-level tasks (e.g. path planning, homing). It is enough to know on which edge the robot is traveling when leaving a node to have fast and large-scale navigation execution. Unlike the geometric maps, they do not require a metric sensor model to convert allothetic data into a common frame reference [Filliat and Meyer, 2003]. The only requirement is to store and recognize the place fingerprint from sensor readings. Therefore, they normally require less memory than metric maps and, hence, have lower complexity. Topological maps solve the global localization problem. Furthermore, they provide a means to solve loop closing [Ho and Newman, 2007] and the kidnapped robot problem [Engelson and McDermott, 1992; Andreasson and Duckett, 2004], because they can deal with the arbitrarily large errors in the robot's odometry. Figure 2.8 shows a loop closing example where appearance-based place recognition identifies similar regions that can close the navigated loop. Finally, they are strongly related to the semantics and provide a means of inserting context into the modeling if object classes are associated with them.

Despite those advantages, topological maps are not problem free. Firstly, they are apparently easier to construct than metric maps, but their autonomous building is difficult. Practically, the decision of when to add a new node to the map is not always clear [Ramisa, 2009]. Secondly, localization with them is still susceptible to general perceptual aliasing and data association problems, which makes node recognition not accurate enough [Buschka, 2005; Guivant et al., 2004]. This second drawback can limit their employment for localization in certain environments, especially in large-scale ones.

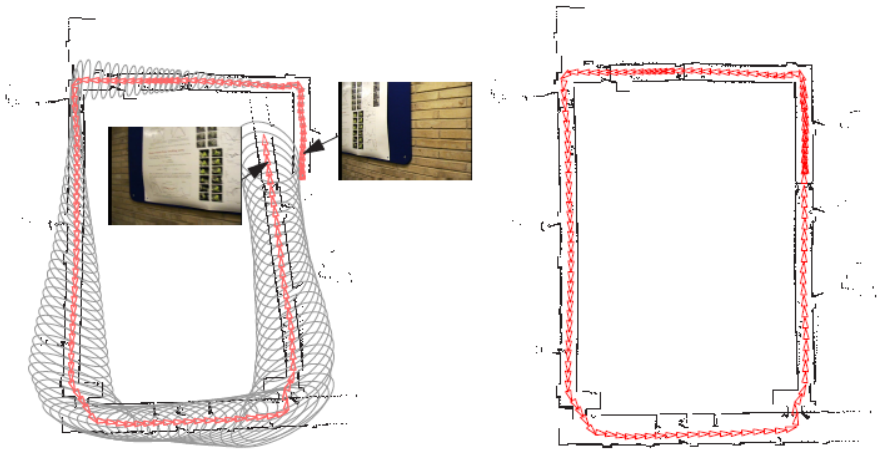


Figure 2.8: Loop closing example [Ho and Newman, 2007]. (a) A snapshot of a SLAM just before loop closing shown in red. Global uncertainty (gray ellipses) increases with the length of the excursion. A poor scan match at the bottom right introduces a small angular error which leads to a large error in pose estimate when the robot returns to starting location (top right). The inset images are the two camera views (current and matched) which are used to close the loop and correct the pose. (b) The corrected map after loop closure detection.

2.3 Localization

Robot localization is the problem of estimating the robot's position in a map. Usually, this estimate cannot be sensed directly in a reliable way. Absolute position sensors like GPS are unsuitable for use in all environments, while reference position sensors like odometers generate large measurement drifts and give indications about the robot's relative position only. Therefore, the robot's position has to be inferred with the help of both an internal map and sensor data. The sensory perceptions are compared to the map in order to correctly locate the robot, or update its position estimate and keep the error or uncertainty bounded.

Table 2.3 shows different criteria for classifying localization approaches. The typical classification is *metric* versus *topological* localization, in accordance with the space representation and the type of employed map [Desouza and Kak, 2002; Bonin-Font et al., 2008]. Figure 2.9 illustrates different position estimates in differently employed spatial representations, and accordingly maps types, for the same environment. The estimated position can be metric in a continuous space, taking the form of a Cartesian location and orientation vector $[x,y,\theta]^T$ as in figure 2.9-b. It can also be metric represented as a bounded area in a discretized metric space as in figure 2.9-c (i.e. cell or neighboring cells in an occupancy grid). Finally, it can be a topological place in a discrete topological map as in figure 2.9-c (i.e. node).

Another criterion for classifying localization approaches is *absolute* versus *relative* [Desouza and Kak, 2002; Bonin-Font et al., 2008; Thrun et al., 2005]. Absolute localization locates the robot without the need of prior knowledge about robot past positions, while in relative localization it is a main requirement. A third classification is *probabilistic* versus non- *probabilistic*. This classification originates from a recent terminology that refers to robots employing Bayesian filters for map building and localization as probabilistic robots [Thrun et al., 2005]. Probabilistic localization is also termed pose tracking.

Table 2.3. Localization approaches classification.

Metric	*versus*	*Topological*
Absolute	*versus*	*Relative*
Probabilistic	*versus*	*Non-probabilistic*

The second localization classification has been also regarded in the literature differently, as a number of increasingly difficult problem instances or situations [Thrun et al., 2001]. In the first situation, the initial robot pose is known, and it is required to track the pose relative to this given position while navigating. Techniques that solve this problem are called tracking or local techniques [Fox et al., 1999], and such situation is referred to as *local localization*, *relative*/*incremental positioning* or *position tracking*. Solution approaches for this situation accommodate the noise in the robot motion. Since the effect of noise is usually small, position tracking methods often assume that the pose error is locally bounded and approximate the pose uncertainty by a unimodal distribution, such as a Gaussian.

In the second situation, no a priori knowledge about the robot starting position is given. Therefore, no assumptions can be made about the boundedness of the pose error and the robot has to localize itself from scratch. This is referred to as *absolute* or *global localization*, and sometimes referred to as the *wakeup-robot problem*. Methods that solve this problem are called global techniques [Fox et al., 1999]. The localization must construct a match between the observations and the expectations as derived from the observed and perceived spaces. Solution approaches include triangulation methods and Bayesian filters that accommodate multiple tracking.

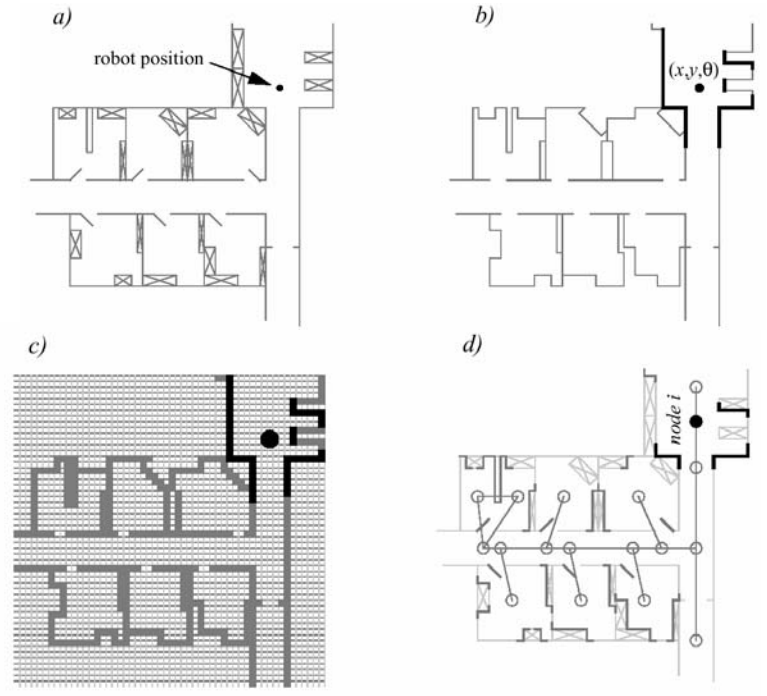


Figure 2.9: Localization classification - metric versus topological: (a) An actual floor map example with the robot's exact position. (b) Continuous metric localization $[x,y,\theta]^T$ (in a metric map). (c) Discrete metric localization (in an occupancy grid). (d) Discrete topological localization (in a topological map).

A third and even more difficult situation deals with the case in which the robot is suddenly transferred or 'kidnapped' to another position without being aware of it. This is identified as a *kidnapped* or *lost robot problem*. The robot doesn't have to be physically teleported, but it is rather a simulation that the robot might have encountered an internal error or reset operation. This problem is critical since the robot might still have the belief that it is in the old location and severe consequences might occur because of that. In this case, the robot has to identify first that it has been kidnapped and then find out its new location. The wake-up robot problem is considered a special case of the kidnapped robot problem in which the robot knows that it has been kidnapped.

Solution approaches developed in the literature for map-based localization are plentiful and cover the mentioned localization classification. A particular implementation depends on the application scenario, environment characteristics and the sensory equipment. In the following subsections, the commonly employed methods for both metric and topological localization of mobile robots are reviewed. Probabilistic and non-probabilistic localization methods are covered in the metric localization subsection.

2.3.1 Metric Localization

In the metric localization problem, it is required to estimate a robot's location and orientation – together called *the pose* – relative to a given coordinate frame. It is assumed that the robot holds a metric map of the environment and it tries to determine its pose in this map. Localization using metric maps is an intensively explored research area. In what follows, solutions are presented from the classification point of view of probabilistic pose tracking and non-probabilistic localization methods.

2.3.1.1 Probabilistic Methods for Pose Tracking

Probabilistic pose tracking methods track the robot's pose using iterative Bayesian filters. Bayesian filters update a belief they maintain about the pose by integrating perceptions and control cues over time. This belief is a conditional probability density over the set of possible poses. The control in this sense is an action indicating that the robot has executed a move. A typical example of control data is the velocity of robot. The perceptions are observations or sensor measurements obtained from a range sensor or a video camera. The example in figure 2.10 illustrates how the integration of data over time helps the robot resolve ambiguity in its position. In the example, the robot cannot decide its location after a single measurement of recognizing the first door. The robot has to move forward, and so its current belief is updated by fusing the previous belief with the current observation. The location estimate for the illustrated example is resolved after two steps.

Figure 2.11 depicts the graphical model for the mobile robot localization problem in terms of a dynamic Bayes network [Thrun et al., 2005]. The robot is given a map of the environment m and the goal is to determine its state x_t (position) at given time t relative to this map, given all the data which the robot recorded or computed up to time t. These correspond to the controls $u_{0:t}$, the environment perceptions $z_{1:t}$, and the previously computed poses $x_{0:t-1}$.

In mathematical terms, the robot's belief about the location $bel(x_t)$ is a posterior in the form of $p(x_t | z_{1:t}, u_{1:t})$. The system starts out with an initial probability distribution $\overline{bel}(x_t)$ of $p(x_t | z_{1:t-1}, u_{1:t-1})$, and modifies it based on recursive Bayesian estimation by iteratively integrating actions and evidences as follows [Thrun et al., 2005]:

$$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1} \tag{2.1}$$

$$bel(x_t) = \eta p(z_t \mid x_t) \overline{bel}(x_t) \tag{2.2}$$

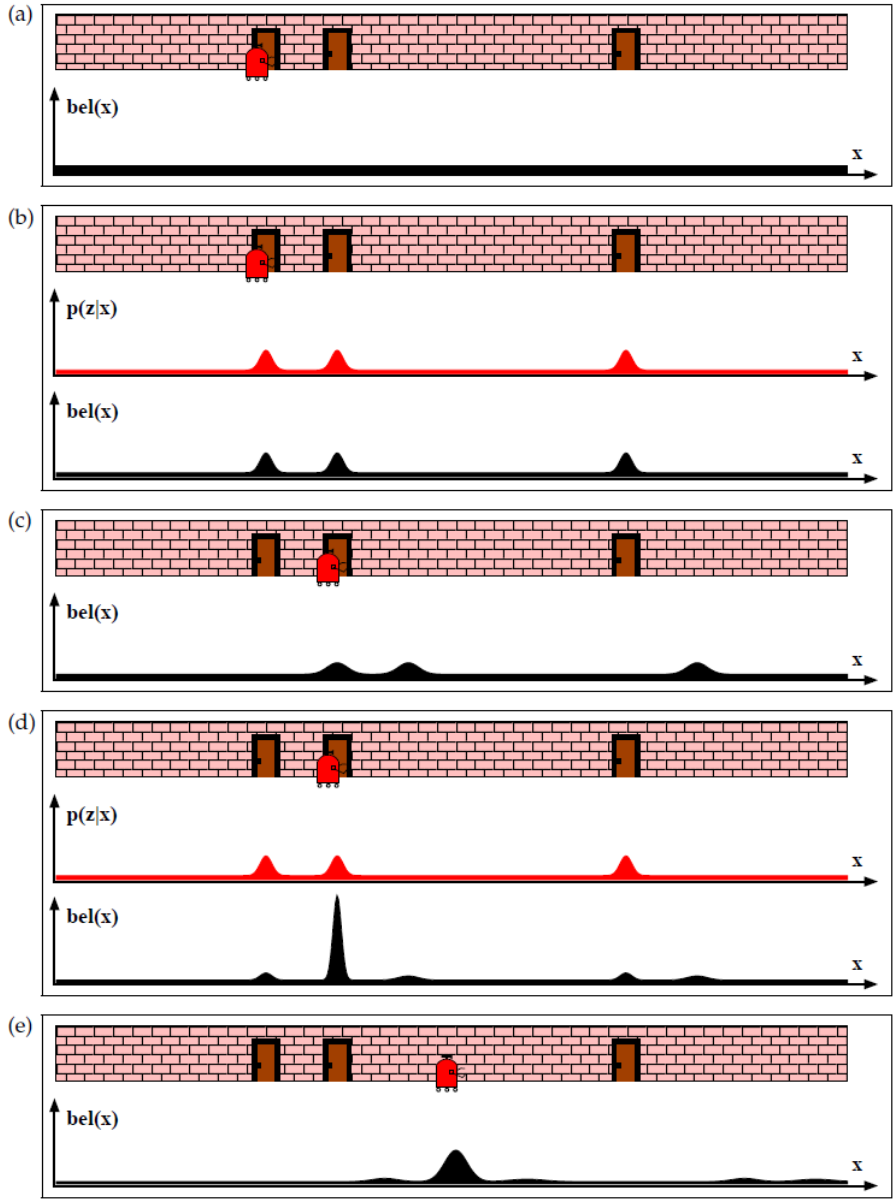


Figure 2.10: Probabilistic global localization (Markov localization) – an illustrative example [Thrun et al., 2005]. Each picture depicts the position of the robot in a corridor along with its current belief bel(x) represented as a probability density function. (b) and (d) additionally depict the observation model p(z|x), which describes the probability of observing a door at the different locations in the hallway. A robot's current belief is updated by fusing the previous belief with the encountered observation.

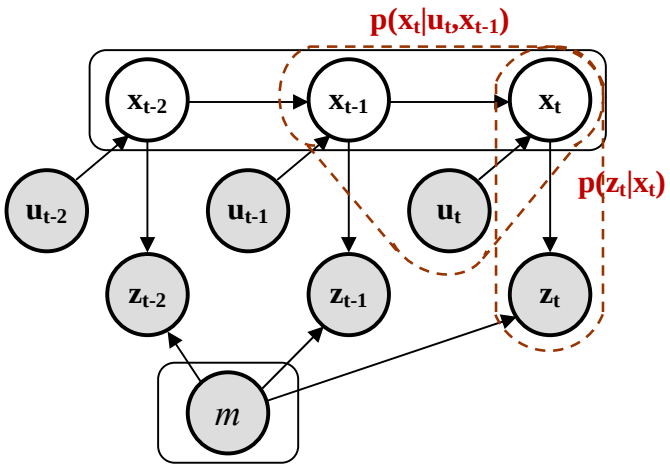


Figure 2.11: Graphical model of mobile robot localization in the form of dynamic Bayes network, which characterizes the evolution of controls, states and measurements. The values of the shaded nodes are given: the map *m*, the measurements z, and the controls u. The goal of localization is to infer the robot pose variable x.

Equation (2.1) is the robot's belief after executing a movement but before an observation is made, and is called the *Prediction Step*. Equation (2.2) is the robot's belief after an observation is made and is called the *Correction* or *Measurement Update Step*. η is the Bayesian normalizing constant relative to the state $p(z_t|\ z_{1:t-1},\ u_{1:t})$. In the literature, the transition density $p(x_t|u_t,x_{t-1})$ is referenced as the *action* or *motion model* and is approximated from the kinematics and dynamics of the robot. It is defined either through a velocity motion model or an odometry motion model. In practice, odometry models are more accurate than velocity models, for most robots do not execute velocity commands with the level of accuracy that can be obtained by measuring the revolution of robot's wheels [Thrun et al., 2005]. It should also be noted that odometry is available only after the motion command has been executed, while velocity commands are available before performing the actual motion. $p(z_t|x_t)$ is referenced as the *observation, sensor* or *measurement model*. Unlike the transition density of the action model, the observation probability density is difficult to compute because of the high dimensionality of measurements.

As (2.1) and (2.2) indicates, Bayesian filters usually simplify the iterative processing by following the Markovian assumption. It assumes that the estimated position is a function of observed and control data between current and previously occupied positions only, and not of the complete data history (first order Markov model). That is to say:

$$p(x_t \mid x_{0:t-1}) = p(x_t \mid x_{t-1}) \tag{2.3}$$

$$p(z_t \mid z_{1:t-1}) = p(z_t \mid z_{t-1}) \tag{2.4}$$

The implementation of a Bayes filter requires a representation for the belief function, the initial belief state, and the motion and sensor models. Implementation techniques differ primarily in the way the belief is represented and how the update of the belief is calculated. Kalman Filters, Markov localization and Monte Carlo localization (MCL) are common techniques for implementing the Bayes filter. Kalman filters implement the belief in continuous form, while Markov localization and MCL implement it in discrete form. The latter techniques are originally derived by Thrun and colleagues [Fox et al., 1999]. They form the main probabilistic framework foundations of many localization methods currently in use.

Early metric probabilistic methods focused on single hypothesis tracking in the metric space using Kalman filters and its variants [Filliat and Meyer, 2003; Crowley, 1989; Leonard and Durrant-Whyte, 1991]. The Kalman filter (KF) smoothes out the effects of noise in the estimated state variable by incorporating more information from reliable data than from unreliable data. It also makes it very easy to combine measurements from different sources (i.e. sensor fusion).

KF [Welch and Bishop, 2006] treats the probabilities ($\overline{bel}(x_0)$, $p(x_t|u_t,x_{t-1})$ and $p(z_t|x_t)$) as parametric Gaussian distributions. It calculates the belief as a single Gaussian function characterized by its mean and covariance, and described by linear state equations. The measurement probability must be linear in its arguments; the same as the state transition probability. In practice, this might not always be true, but it does allow the KF to efficiently make its calculations [Fox et al., 1999]. The KF computes the a posteriori belief as the linear combination of a priori estimate and a weighted difference between the actual measurement and its prediction. This discrepancy (between predicted and actual measurements) is called the innovation, and is fused with a blending factor in order to update the a posteriori. The blending factor is called the Kalman gain, and is calculated based on minimizing the a posteriori error covariance. The Extended Kalman Filter (EKF) is a variant of the classical filter that disregards the linearity assumption to deal with non-linear state and observation models. The non-linear functions are approximated through first order Taylor expansion.

Different features are used with Kalman filters for robot localization. For example in [Crowley, 1989], lines are extracted from sonar data. Matching observations to a local model is applied as the correction step of a linear KF for the estimated robot position at the time the observation was made. [Leonard and Durrant-Whyte, 1991] matches geometric beacons (walls and corners) extracted from sonar scans against those predicted from a geometric feature map in an EKF. Similarly, an EKF is employed in [Bonnifait and Garcia, 1996] to fuse odometry and angular measurements to known landmarks (light sources which are detected using a CCD camera) together.

Kalman filters allow high accuracy with low complexity, since the belief distribution is simple. Only one single pose is considered, and whenever an action is performed or an observation is made, the estimate is updated given this only hypothesis. Computing the posterior means computing only new mean and covariance from the old data, using the action and sensor readings. Hence, Kalman filters possess this advantage that the representation is simple and the update computations are cheap. Their disadvantage is that position tracking might be lost if the measurements are in some way ambiguous. The initial position must be also known. Their solution remains unimodal, which means they are limited to solving local localization only, and cannot handle global or kidnapped situations. The big difference between the estimate and the real position can drive the system to completely wrong position, from which the system is unable to recover.

To overcome such problem of having to know the initial robot position and the fatal error that can occur if the position is lost due to a wrong estimate at some time, simultaneous tracking of multiple position estimates has been proposed. This method is simply an extension of the single hypotheses of Kalman filters. Multiple Filters are employed to track several position hypotheses [Piasecki, 1995; Jensfelt and Kristensen, 2001]. The number of hypotheses adapts to the uncertainty of localization. They are induced on those positions that have generated the current perception. The method provides the missing robustness against lost situations. However, it becomes infeasible if the robot has to re-localize itself during motion due to the high computational complexity.

Other types of filters can solve the previous problems of KF, such as topological graphs, grids or particle filters. These filters are based on discrete belief representation. Both topological graphs and grids are similar in dealing with a discrete representation for the

space, which at the same time represents the state to be estimated. Topological grids use a spatial discretized grid to compute a discrete approximation of a probability distribution over all possible poses. When the location space is discretized, the integrals in equation (2.1) become sums and the belief over this space can be explicitly computed and stored. The grid is similar to the occupancy grid, but with every cell preserving a belief of how likely the robot is located in that cell, instead of an occupancy value. Cells are updated using the Bayesian updating. Since no specific features are stored in the grid, the measurement model often uses raw sensor data instead of extracting features from the data. The topological graph proceeds the same way as the topological grid. Localization which employs both filters is called Markov localization [Fox et al., 1998; 1999; Kosecka and Li, 2004]. Markov localization has proven to be both accurate and robust. It is mutli-modal; thus handles the global uncertainty well. It is precise when a small number of cells or nodes is used, but suffers from the heavy computation burden with large grids or topology. Some extensions were introduced in [Burgard et al., 1998] to overcome this problem.

The idea of representing a discrete localization belief with a probability distribution function is also utilized in MCL [Dellaert et al., 1999]. MCL proposes an improvement over the previous methods by employing sampling-based methods (particle filter). Such methods represent arbitrary distributions which offer a trade-off between precision and real-time performance. In a particle filter, the probability distribution over the robot location is represented by a set of weighted samples (importance factors), whose likelihood is updated with each control and perception information. Estimation of the state is given through the sample with the largest likelihood. The number of calculations is reduced compared to grid-based Markov localization. Adaptation of the samples size has been introduced in [Kwok et al., 2003] to reduce the computational cost. Several of the recent applications use MCL with range sensors for localization [Dellaert et al., 1999; Thrun et al., 2001; Kwok et al., 2003; Laaksonen, 2007; Grisetti et al., 2007], while a few still adopt vision sensors [Wolf et al., 2005; Menegatti et al., 2006; Bennewitz et al., 2006]. Figure 2.12 shows an example of multi-hypothesis tracking that resolves global localization using MCL.

The main characteristics of the discussed probabilistic localization methods are summarized in Table 2.4. Generally, methods using features and landmarks permit handling environment dynamics as they reject much of the raw data. The accuracy of the method is

(a) (b) (c)

Figure 2.12: Example of multi-hypothesis tracking using MCL. (a) Samples are distributed randomly over possible positions. (b) As the robot moves, only few hypotheses are tracked where ambiguities due to environment similarities exist. (c) Samples condense around a single position hypothesis [Dellaert et al., 1999].

usually bounded to the quality of the sensor data and not to the method itself. Robustness, in the presented sense, is defined as the ability to avoid lost robot situations (i.e. Multi-hypothesis Tracking (MHT)). The evaluation of MCL is considered a special case, because in order to be practical, the number of particles have to be reduced making it relatively vulnerable to lost situations. As can be seen from the table, there is no method fulfilling the complete list of the properties. The probabilistic approach, however, gains a large popularity for many applications. From a practical view, a specific implementation method can be selected based on the application, environment and sensor requirements.

Table 2.4. Comparison between the different methods of probabilistic localization [Thrun, 2002].

	EKF	MHT	Coarse (topo-logical) grid	Coarse (metric) grid	MCL
Measurements	Landmarks	Landmarks	Landmarks	Raw measurements	Raw measurements
Measurement noise	Gaussian	Gaussian	Any	Any	Any
Posteriors	Gaussian	mixture of Gaussians	Histogram	Histogram	Particles
Efficiency (memory)	++	++	+	–	+
Efficiency (time)	++	+	+	–	+
Ease of implementation	+	–	+	–	++
Resolution	++	++	–	+	+
Robustness	–	+	+	++	++
Global localization	No	Yes	Yes	Yes	Yes

2.3.1.2 Non-probabilistic Methods

Other localization methods are non-probabilistic. These include: *direct scan matching* and *triangulation*. The first method falls into two sets depending on the information used; raw data or features. Dense raw scans have been used directly to infer the robot pose, without explicitly deciding what constitutes a landmark. The pose is inferred by matching the sensor readings against a reference surface map. The discrepancy between them is minimized in an optimization problem by finding the best alignment for the local range measurements with the maximum overlap [Lu and Milios, 1997; Biber and Straßer, 2003; Diosi and Kleeman, 2005]. The other set of scan matching is feature-based. It extracts distinguishing features from the range data and uses them for calculating the alignment of the scans [Lingemann et al., 2005]. Direct scan matching is more popular than feature-based matching, because the strict constraint of the existence of a certain type of geometric feature in the environment is eliminated. However, it operates with memory-intensive maps compared to feature-based matching, since the maps consist of raw measurements recorded from the reference positions.

Scan matching can be performed locally or globally. Local scan matching [Lu and Milios, 1997; Sandberg et al., 2009] performs the matching with an initial pose estimate usually acquired from the odometry. Based on this estimate, the search is limited to small perturbations of the sensor scans for the area consistent with the current robot position, while the areas distantly apart are discarded. Contrarily, global scan matching [Tomono, 2004] aligns the current scan with respect to a map in the form of a database of scans. An initial pose estimate need not to be assigned, and the robot can be globally and precisely localized given accurate inputs. Though local scan matching reduces the required number of computations, it sacrifices the ability to globally localize the robot.

Range scan matching has also been emulated through omnidirectional vision in [Menegatti et al., 2006]. Floor images are searched for color transitions and range measurements are obtained and matched to the prior map in MCL implementation.

The second non-probabilistic localization method is based on landmark triangulation. Sensor readings are analyzed for the existence of landmarks around. Given that these objects are registered in the map, and by incorporating distance (range) or directions (azimuth angles) to them, the absolute position of the robot can be inferred.

As outlined before, landmarks are environment reference points that a robot can detect and use to calculate its pose. A landmark can be a single feature or a combination of features that refer to a higher-level abstraction or an object. It can also be natural or artificial. Examples for natural landmarks are doors, windows and ceiling lights in indoor environments, whereas roads, buildings, trees, sidewalks, and traffic signs are outdoor environment candidates. Vertical lines and point features constitute simpler abstractions of natural landmarks suitable for triangulation [Krotkov, 1989; Goncalves et al., 2005].

Artificial landmarks are human-engineered structures introduced to the environment. They can be passive or active. Passive artificial landmarks resemble natural landmarks, except that they are human-made constructions. Active landmarks are based on a different technology. They are radio transmitting objects that send out signals as a kind of location information. Sometimes, they are called beacons. Beacons are accurate to locate when their signals are strong enough, but structuring and maintaining the environment with them may be costly. Few constraints and problems still exist with those technologies [Singhal, 1997]. A sufficient number of them must be visible from all possible robot positions. In practice, beacons cannot send out their signals in all directions (i.e. omnidirectional transmission), and thus cannot be seen from all places. Additionally, the transmitting of active signals can get disturbed by atmospheric and geographic influences while going from the sender to the receiver. Disturbances like refractions and reflections result in incorrect measurements, and hence triangulation is less robust.

The choice between the kinds of landmarks to use is arbitrary. Each kind reports its pros and cons. Artificial landmarks require a change to the environment, which may not be preferable. Moreover, they may not be feasible in very large environments. They are recommended for use in structured and highly similar environments which do not change very much. Hence, they are easier to locate with more accuracy. Natural landmarks eliminate the burden of changing the environment by already being part of it. They do not have to be engineered in any manner. In this sense, using natural landmarks seems easier. They are recommended for use when the environment possesses suitable amount of details, such that existing variation provides a natural means for recognition, without caring about where to introduce artificial aids. They are also highly recommended for outdoor environments. Compared to artificial beacons, natural landmarks improve over them in terms of robustness

because of their passivity. Nevertheless, the computational complexity of recognizing them is higher and the reliability of their recognition is also lower in similar environments.

Triangulation is the common term for absolute positioning using landmarks, but more precisely, there exist two techniques for this purpose: triangulation and trilateration [Calabrese and Indiveri, 2005]. Triangulation techniques use angular measurements, and sometimes combined with distances. On the other hand, trilateration techniques use distance measurements only [Siadat and Vialle, 2002]. A GPS, for example, uses trilateration to determine latitude, longitude and elevation through the time of fight information of uniquely coded radio signals sent from satellites. Any triangulation problem can be reduced to a trilateration problem by computing the intersection circles subscribing the landmarks (at least three circles for the 2D plane). An advantage of triangulation over trilateration is that the complete pose can be recovered (i.e. orientation in addition to the position). That is why triangulation techniques are more commonly applied than trilateration. Landmark triangulation is considered a robust, accurate, flexible and widely-used method for absolute localization [Esteves et al., 2003].

In triangulation, the angles and/or distances are used to derive the full pose of the robot through geometric and trigonometric relationships. Usually, two landmarks are required to derive the position if the orientation is known. Otherwise, at least three landmarks are required. Figure 2.13-a illustrates that the robot pose *P* is constrained to the arc of a circle, given only the angle measured between two distinguishable landmarks. When an additional landmark is observed, the pose is constrained to a single point lying at the intersection of the two circles (figure 2.13-b); provided that no two landmarks are coincident (i.e. all landmarks are visible to the robot). When four or more landmarks are observed, the triangulation solution is overdetermined and can be solved using least-squares analysis [Betke and Gurvits, 1997; Siadat and Vialle, 2002]. Overdetermined solutions help minimizing the uncertainty bound arising from non-accurate sensor measurements. Singularity in the triangulation of figure 2.13 will arise if the robot point *P* lies on the circumference defined by the three landmarks. This is because both circumferences are the same and therefore their intersection cannot be determined.

One of the earliest triangulation applications is that of [Atiya and Hager, 1993]. It uses vertical lines whose positions are assumed to be known. In a stereo, the robot recognizes

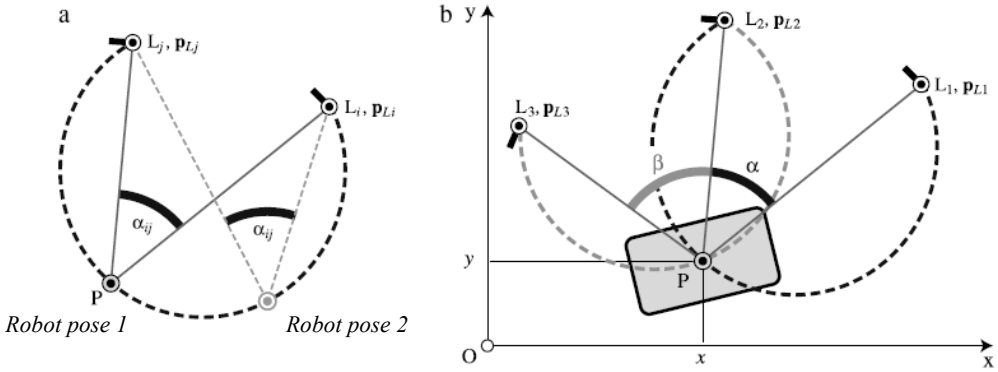


Figure 2.13: Triangulation: constraints of pose given the bearings. (a) Two landmarks: the knowledge of angle $\alpha_{ij} = \theta_j - \theta_i$ constrains the robot position to be on a circular arc through landmarks Li and Lj. (b) Angles α and β define two circular arcs whose intersection is the pose P [Font-Llagunes, Batlle, 2009].

pairtriple landmarks, with sufficient length and angle attributes to set up correspondence between landmarks in the environment and pixels in the camera. Once correspondences are established, finding the absolute position of the robot simply becomes the exercise of triangulation. Localization is executed in real-time. The triangulation implementation is divided between range-camera sensors and artificial-natural landmarks employment. Laser systems have been used in [Hernandez et al., 2003; Font and Batlle, 2006], while projective geometry in conjunction with a pinhole camera model has been adopted in [Yuen and MacDonald, 2005; Welch et al., 1991]. The latter has the advantage that it can localize autonomous vehicles in 3 DOF. Human artificial landmarks were constructed and recognized using vision in [Briggs et al., 2000; Jang et al., 2005], while in [Jang et al., 2003], a combination of artificial and natural landmarks were used together for a combined topological and metric localization. Artificial beacons were also used on a wide scale such as in [Pierlot and Droogenbroeck, 2009; Lee and Song, 2007].

Several methods exist for solving the triangulation problem. Four common numerical and closed-form solutions are provided in [Cohen and Koss, 1992], and which have been applied later in different works: Geometric Triangulation [Easton and Cameron, 2006], Iterative search, Newton-Raphson iterative search [Siadat and Vialle, 2002] and Geometric Circle Intersection [Font-Llagunes and Batlle, 2009]. Both of the geometric methods require the landmarks to be ordered in a given fixed notation. This constraint has been relaxed through a generalized geometric triangulation algorithm in [Esteves et al., 2003], which improves on the one introduced in [Cohen and Koss, 1992].

Triangulation error analysis has been conducted in some studies, where the landmark noise (bearing errors) is propagated to the estimated pose [Easton and Cameron, 2006; Madsen et al., 1997; Shoval and Sinriech, 2001; Esteves et al., 2006]. Different regions are identified where triangulation is robust and accurate, while in other regions large errors are exposed. Results comply with [Kelly, 2003], which states, without quantification, that the error in the pose will increase as the distance between the robot and landmarks increases, and that the error size will decrease with the increase of the sine of the angle between the two landmarks. Error analysis helps designing accurate triangulation through optimal selection of landmarks that minimizes the pose uncertainty. In [Easton and Cameron, 2006], the effect of landmarks structure for the accuracy of triangulation is studied. For example, it has been shown that accuracy is higher if the landmarks are spaced close to an equilateral triangle, while lower if they lie on a straight line (a singularity of the triangulation). In [Madsen and Andersen, 1998], some guidelines were laid regarding a good selection of landmarks.

2.3.2 Topological Localization

Localization in topological maps is basically to identify in which node or edge the robot is. In contrast to metric localization, topological localization is a direct position inference done via place matching to recognize the node. The detection of edge traversal is recognized through the odometry or the robot's actions. To infer the node accurately, most topological localization implementations use distinctive fingerprints to characterize the nodes. Normally, those fingerprints are feature-based descriptors. Distances and angles at the branching inter-connections of node can account for the descriptor too [Choi et al., 2002; Choset and Nagatani, 2001]. The node inference, in general, is still liable to perceptual aliasing despite this rich characterization. Aliasing occurs due to noisy measurements, reduced data by feature extraction algorithms, or similar data because the environment itself has repeated structures (e.g. all corridors will look the same in a maze environment).

Topological localization is efficiently presented by employing vision sensor-based features than range sensor-based features alone. This is because perceptual aliasing is less in the vision measurement space than the range measurement space. Several vision-based approaches have been presented, which mainly differ in the way the scene is perceived. Examples for used cues are image color histograms [Ulrich and Nourbakhsh, 2000; Blaer and

Allen, 2002], eigenspace representation [Artac, 2002], and frequency filter responses [Menegatti et al., 2004].

The first original study for topological localization and place recognition is the work done in [Ulrich and Nourbakhsh, 2000]. In this work, the robot is led through an indoor environment taking pictures with a panoramic CCD camera, and images are classified in real-time based on nearest neighbor learning, image histogram matching, and a simple voting scheme. It is claimed that the color images contain enough information that no additional range sensor data is needed. In [Lamon et al. 2003], an interesting 'fingerprint' concept is introduced for visual localization by recovering a circular list of combined features from 360º color images, where the ordering of the set matches the relative ordering of features around the robot. Part of the information is coded in the nature of the features (vertical edge, corner or color patch), and another in the order of the list. A minimum energy optimization algorithm based on cost matrix is used for the fingerprint matching.

Other significant work employs regular cameras for the image acquisition. In [Jung and Kim, 2005], a webcam is used to extract a Gaussian model of texture distribution from a steerable pyramid using wavelet decomposition. Kullback-Leibler divergence is used for image matching. The success rate of place recognition is recorded to be 87% on average. A major contribution presented in [Pronobis et al., 2006] is a vision recognition approach for five-place office environment under varying illumination conditions. Receptive field histograms are used for feature extraction, with a support vector machine classifier. The approach achieved a classification rate of 74.5-84.6%, with remarkable recognition time. In [Mozos et al., 2007], an integrated vision-laser sensing succeeded in extracting geometric features from an indoor environment and classifying it into door, corridor, hallway, office and kitchen classes. The incorporation of the spatial dependencies, using a Markov model with AdaBoost classifier, managed to greatly distinguish between the different classes.

Holistic approaches appear to work efficiently as well for topological recognition. In these approaches, recognition is accomplished based on a general appearance of the scene. This general appearance is often captured through spectral information. These approaches show general efficiency, but are known to be sensitive against rotation and translation transformations which may reflect scene dynamics. In [Oliva and Torralba, 2001], the use of shape is argued to be an underlying stable spatial structure that presumably exists within all

scenes. A gist descriptor is proposed consisting of Discrete Fourier Transform (DFT) coefficients, which are compressed by the Karhunen-Loeve Transform (KLT). The method shows satisfactory results when applied to outdoor environments. The descriptor's performance, however, dropped dramatically when applied to indoor environment categories. Another gist, CENTRIST, is proposed in [Wu and Rehg, 2010] claiming to perform efficiently both indoors and outdoors. The descriptor is based on a window operation with a Census Transform that encodes and compresses information in local patches.

Histograms of various image properties, such as color and image derivatives, have been widely applied for place recognition. However, after the SIFT feature descriptor [Lowe, 2004] was introduced, it became popular and dominated several vision applications including localization systems [Kosecka and Li, 2004; Ledwich and Williams, 2004]. The features have been incorporated in direct topological matching using several distance measures and classifiers [Sabatta, 2008; Ullah et al., 2008], and sometimes fused with motion models in MCL [Bennewitz et al., 2006]. Different comparison studies of SIFT against other local and global descriptors were conducted, in which the SIFT's accuracy was high and stable [Mikolajczyk and Schmid, 2003; Ramisa et al., 2008; Valgren and Lilienthal, 2007]. For faster execution of SIFT, some work suggested the elimination of the rotation invariance step from the descriptor's processing [Ledwich and Williams, 2004], or the use of only persistent features which are robustly detected [Sabatta, 2008]. The latter can limit the total amount of information required to map an environment by 70% compared to other methods that store SIFT descriptors directly. Another local descriptor, SURF [Bay et al., 2008], has also proved in few works to be as good competitor as SIFT for mapping and localization [Valgren, 2007].

2.4 Hybrid Map Building and Localization

In the previous sections, the classical classification for map building and localization is introduced. Metric maps allow the robot to compute optimal paths and perform accurate localization. The maps are, however, hard to create and maintain due to the inaccuracies in the robot motion and sensing. Moreover, they involve heavy computational costs affecting performance and scalability in large environments, and do not interface well with symbolic problem solvers and humans [Buschka and Saffiotti, 2004; Thrun and Bücken, 1996]. In contrast, topological maps scale better to large environments and interface more naturally

with symbolic systems and humans. However, they allow only coarse localization and suboptimal path planning. Besides, they have difficulty in distinguishing between different places when additional metric information is not employed. Table 2.5 compares the metric versus topological paradigms. In a trail to combine the precision of the former with the scalability of the latter, the hybrid mapping and localization has emerged integrating both paradigms. The field is new and actively growing in research. It targets the need for systems that provide dual functionalities, with faster and scalable geometric localization than the classical metric approaches.

Table 2.5. Comparing topological to metric map building and localization.

	Metric	**Topological**
Resolution	High	Low
Computational time	High	Low
Memory utilization	High	Low
Maintenance(map updates)	High	Low
Scalability	Low	High
Sensitivity to noise	More	Less
Perceptual aliasing	exists	exists
Map size	Larger	Smaller

Although a hybrid map consists generally of any combination of different maps, the hybridization is most often performed using topological and metric maps. A semantic map is also a term related to the design of hybrid maps. Local metric maps can be connected to a global topological map which can be described by a third semantic layer comprising information about rooms and objects, such as kitchen, bedroom, bed, sofa, etc [Galindo et al., 2005]. Most of the presented work, if not all, performs the 'hybridization' in the form of a hierarchical structure consisting of two levels, incorporating two maps and two localization modules. In this view, a precise frame structure for the hybrid map is defined; the topological map occupies the higher level of the hierarchy, while the metric map occupies the lower one. The initial motivating work in the area of hybrid map building and localization is [Thrun, 1998]. In this work, the grid-based approaches are compared to the topological approaches, and it is concluded that a combination of both approaches gives the best results with respect to computational complexity, preciseness and scalability. Obviously, the merit of the hybrid

approach is seen in the hierarchical processing of localization which handles the geometric localization problem in economic means.

Hierarchical map building and localization represent and process information in at least two separate levels. The hierarchy moves top-down in accordance with the resolution of information; from course to fine scale. In addition to providing an organized structure for data, the goal of hierarchical processing approaches is to reduce the set of candidates towards a solution, and hence, the high computations are evaluated only on a minimal subset rather than the entire reference set. Therefore, hierarchical localization frameworks provide much better metric localization performance in large-scale environments. The initial topological localization shrinks the related metric search space either to a single space or a distribution over it, providing less space and time complexities.

General hybrid maps preserve both the topological and metric information generated in a top-down or bottom-up way. In the top-down case, the map structure is in the form of a global topological map that connects several local metric maps. Usually, the topological map is initially constructed, and next linked to a set of detailed metric maps for each topological node in an obvious hierarchical fashion. An example of the top-down structure is [Tomatis et al., 2003]. It uses corners and openings as topological landmarks and lines as metric ones, both recognized by means of a laser ranger. EKF and POMDP are used for metric and topological localization respectively. When switching from the topological to the metric map, the EKF has to be initialized. For this purpose, a detectable metric feature (a door) between both maps allows knowing when to begin the switch and gives an approximation of the robot position in the local metric map. In [Tully et al., 2007], a hierarchical atlas has been constructed linking a high-level topological graph (GVG) to a set of lower level feature-based metric submaps. A discrete Bayes filter and a KF are employed together to localize the robot at the two levels. Similar work is done in [Lisien et al., 2003] in a SLAM approach.

The second bottom-up structure has another perspective. One can say it is of interest to decide how to distribute the environment into and between local maps. Therefore, obvious map hierarchy might be missing, since the idea is based on grouping. It can be seen only in the localization as it is top-down executed. This group coincides with the previously presented approach of generating topological maps from CADs or other forms of metric maps. The main idea is to cut the high-resolution map into disjoint areas. That is why a global metric or high-

resolution map is first constructed from which the topological map is generated on the top of it by use of splitting measure. A relevant advantage for this structure is that the topological map preserves consistency with the high-resolution metric representation. This allows an easy and consistent switch between the two maps whenever a specific resolution is required. Another advantage is that the topological nodes are created automatically.

In [Blanco et al., 2006], the recursive partitioning of the sensed space based on overlaps generates the topological map from the metric one obtained by SLAM with laser scans. It is assumed that the absolute poses, from where the different scans were taken, are available. The method then considers the space sensed in each observation as a node of a graph, whose arcs represent the sensed-space overlap between two observations. Recursive cuts of this graph, based on intra-group cohesion metric, produce groups of strongly connected nodes. In [Thrun and Bücken, 1996], a topological map is extracted from a grid map using the Voronoi method in the classical way. The grid map is generated by training an artificial neural network that maps sonar measurements into occupancy values.

It appears normal to extract a topological map from a metric one. Though the reverse seems to be odd, it has been carried out in [Duckett and Saffiotti, 2000] to design a hierarchical mapping and localization system. The technique is offline-based, which makes use of metric information in the topological map together with a relaxation method in order to maintain the geometric consistency.

Other than employing an explicit hybrid map, place information has been fused with probabilistic filters to account for hierarchical localization. In [Wolf et al., 2005], an image retrieval system based on invariant features is used as the correction step in a MCL. Similarity values generated from the retrieval system are used to update the weights of samples, together with a visibility region determined from a pre-computed and stored occupancy grid map. [Menegatti et al., 2003a] uses the same idea without the visibility region, and with less number of images acquired from an omnidirectional sensor. Their retrieval system uses Fourier Transform to extract features instead. In [Murillo et al., 2007b], the topological localization first computes the robot position relative to a set of reference images. The reference images are identified through radial lines, which are next fed into a three-view omnidirectional vision system (trifocal tensor) that calculates the bearings and identifies the metric pose.

Different abstractions of features have established a third form of hierarchy towards localization [Courbon et al., 2008; Menegatti et al., 2003b; Wang et al., 2006]. Those methods start with coarse and fast data matching, such as matching using global features. Retrieved candidates enter a second matching phase using another set of data, such as local features to find a single best solution at the end. Good compromise of accuracy, memory and computational cost has been achieved by those methods. High-density data set is, however, an essential requirement for high resolution localization, which is a critical disadvantage.

In terms of hybrid localization methods, the work of [Baltzakis and Trahanias, 2002] can be mentioned. It presents a hybrid technique for localization only, and regards efficiency but not scalability as the previous approaches. The technique uses a switching state-space model employing a probabilistic KF and the HMM. Line segments and corner points are extracted from laser range data. The features are tracked through multiple Kalman trackers, while allowing the probabilistic relations among the multiple hypotheses to be handled by the discrete Markovian dynamics, where duplicate hypotheses are merged and improbable hypotheses are removed.

The advantages of the hybrid map building and hierarchical localization can be summarized as follows: (i) They provide both coarse and fine resolutions for a robot pose in a single framework; (ii) they translate the need to represent and reason about information at different levels of abstractions at the same time; (iii) they provide a sensor fusion approach if the topological space uses a different feature set than that used in the metric space. This would lead to a more accurate position estimate; (iv) performance scales much better for large environments. The hybrid framework additionally provides sound advantages from the topological mapping, such as: (v) Introducing context and semantics for cognitive perception; (vi) solving loop closing and recovering from serious errors of lost or kidnapped robot situations. A crucial and critical issue in those frameworks, however, exists. The accuracy of the metric localization is bound to the topological accuracy. A wrong topological estimation can lead to severe coincidences in the metric space, at least in regard to safety factors.

2.5 Summary

Many potential solutions exist for robotic map building and localization. Basically, they are categorized into two paradigms: metric and topological. Robotic map building in the context

of the late 1980's was primarily metric, meaning that solutions used geometric features or grids to represent the environment. Those solutions provide high-resolution position estimate but on account of high computation requirements. Topological maps, where only places and connections are stored in a graph representation, have been identified in the literature as compact, efficient and scalable representation but on account of the resolution.

Localization solutions are rather categorized on another relevant dimension; that is relative and absolute localization. Relative localization uses relative measurements, such as odometry and inertial navigation fused with Kalman filters. Absolute localization uses probabilistic multi-hypotheses tracking filters such as Monte Carlo Localization, or employs active beacons, artificial/natural landmarks and dense scans in non-probabilistic methods like triangulation and model matching techniques.

Though several environment map building and localization solutions are afforded for the different applications, some challenging problems exist which influence their performances. One group of challenges concerns the perception process, which in turn influence the localization accuracy. The group includes environment perceptual aliasing and data association problems. To minimize the severe effect of those problems, a different concept for modeling the operating environment should be regarded. Another group of challenges concerns the techniques used for the solution in their own, regarding their computational performance and scalability. The hybrid map building and localization paradigm emerged recently to provide a promising solution for the latter challenge group. It combines both the metric and topological paradigms to build computationally efficient and scalable metric localization solutions. In those solutions, the topological accuracy cannot be that easily overlooked.

Chapter 3

Preliminaries to Environment Modeling

This chapter highlights some preliminaries for the solutions presented by this work. It first discusses some general, but basic concepts for environment modeling and perception. Next, the different exteroceptive sensors which form the candidate pool for the proposed work are introduced. The advantages and disadvantages of each sensor and its suitability to the type of environment in general are given. The possible environment modeling approaches and feature extraction are also presented. Afterwards, attention is given to the explanation of local feature extractors and the basic foundations of the information theory. The chapter closes with a short summary highlighting the foundation elements of the proposed work.

3.1 Environment Modeling[1] and Perception

General environment modeling is the process of environment data interpretation and representation for a specific purpose. For the navigation purpose, it is a synonym for cartography or map building and is applied through an interaction between the agent and its surroundings by means of its exteroperceptive sensors.

[1] In the scope of this work, environment modeling is the processing adopted to generate robotic navigating maps, which includes other methodologies rather than using the feature extraction methodology only. Hence, it is a more general term than environment map building.

Maps are descriptive models that work on a higher level of abstraction than state variables models or systems of differential equations. As has been introduced in the related work of chapter two, maps take different forms and are utilized for different tasks. For instance, a topological graph can be employed for path planning, while a list of distinguished objects can be used for local navigation. Similarly, a color map can be applied for workspace identification, and an ultra-sonic map for position update. Practically, there is hardly a single environment model that can be used for all purposes for complex systems. There is also no single model that is capable of capturing all the properties required. Furthermore, there is rarely one model that can map all kinds of environments [Badreddin, 1997]. Nevertheless, investigating efforts usually try to fit a specific model to the environment with regard to some application constraints and the required efficiency.

Perception is an elementary and integral part of the environment modeling, and the employed sensor plays an essential role in the system's behavior. To understand the value of sensors and perception, we can as an example imagine how difficult the task is to explain the red color of an apple for a lifelong blind person. Hence, choosing the proper sensor for the environment modeling is not a trivial issue.

Perception starts by acquiring raw data from the sensors. Raw data can be used directly, but they deliver low information. Therefore, the data are often processed at different levels of perceptual abstraction, and the result is the so-called *"features"*. Features are the most relevant information extracted from the original data according to a certain filter with respect to a lower dimensionality space. Different feature abstractions exist; from primitive color, intensity, or distinct textural patterns, to high-level geometric features such as lines, edges or corners, and moving to complex and sophisticated object models. Normally, the choice of a feature abstraction is a function of the available sensor and the application environment.

A model building is much more than a feature extraction process. From a generative point of view, it is decomposed into the following steps [Badreddin, 1997]:

- Determining *"attractive"* features from which the model, or part of it, is to be built, i.e., *unsupervised learning* of some features.
- *Supervised learning* of prescribed features.

– Establishing the topological relationship between features to deliver the structure which combines the features into a complete model.

As can be deduced, *learning* appears to be an important factor in the model building process. Usually, the lack of explicit knowledge and the probable high-dimensional data spaces suggest learning as a strategic way to construct efficient solution-models. Similarly from the artificial intelligence field perspective, it is viewed as the basic strategy for building intelligent models [Han and Kamber, 2006]. Another considerable factor which cannot be ignored is that learning provides a means of robustness by handling the external noise effects and uncertain sensing and perception.

Being engaged in lots of work, the geometric model represents the most widely applied modeling approach for environment modeling. A complex environment, however, can be better described by a set of layers, each capturing a particular class of features. A mapping of one feature class simply corresponds to the kind of sensor used to detect this feature. For example, besides the geometric layer that describes the general shape of object, another layer can describe the physical properties, such as temperature or conductivity. An example of this layered concept is given in table 3.1.

Table 3.1. Example of classes and their features [Badreddin, 1997].

Class	**Features**
Ultrasonic	Range, Range-image, Intensity, …
Passive Vision	Grey-level, Color, Contour, …
Active Vision	3D Range-image, Intensity, …
Infrared	Temperature, Range, Reflectivity, …
Tactile	Force, Torque, Electrical Conductivity, …
Geometry	Edge, Corner, Plane, …
Objects	Cylinder, Cube, Table, Window, …
Microphone	Sound frequency, …

Moreover, the environment can be described in terms of information-theoretic quantities (e.g. conditional or unconditional entropy or transmission). These quantities provide measures for the detectability and distinguishability of environment objects and features. Usually, the question of which features to employ in the model is not a directly-answered question. It

depends on the purpose for which the model is intended, the available sensors, the type of environment, and the permissible space and time complexities. An information-theoretic modeling for the environment can count as an additional solution to answering this question.

3.2 Sensors

As mentioned in the previous chapter, robot sensors can be distinguished into two types: idiothetic (proprioceptive) and allothetic (exteroceptive). This chapter is concerned with the exteroceptive type, which makes more relevance in the environment model building problem.

Exteroceptive sensors are a major and vital technology that proliferates very fast. Noise and uncertainties remain an inherent part of all sensors. Without exceptions, all measurements provided by sensors are uncertain by nature. Therefore, it is recommended that the uncertainties are isolated and modeled by knowing the physical characteristics of the sensor before errors propagate in the system application levels. Probabilistic and possibilistic approaches can model uncertainties. They are more reliable than relying only on the first measurement moment. In what follows, a variety of sensors suiting the presented work and implementation, together with the cons and pros and suitability for the environment, is introduced.

3.2.1 Range Sensors

Range sensors are used in almost every robotic platform. In the 1970's and 1980's, *ultrasonic sensors* were the popular perception system for acquiring distance information. They are still employed up till now, primarily because of their low cost and easy integration. Like most other range finders, they are based on the time-of-flight principle, where the range distance is estimated by measuring the time a beam requires to hit an obstacle and reflect back to the transmitter. Their disadvantage is their large beam angle, which accounts to a low angle resolution.

Infrared sensors overcome this problem, but they do not extend for longer distances. Besides, as they emit infrared light, they do not function accurately outdoors or even indoors if there is direct and sometimes indirect sunlight. In the last decade, *laser range finders and scanners* gained a large acceptance. They are engaged in a large number of applications mainly because of the high accuracy they can achieve. For example, they have been used extensively in 3D object modeling and recognition. The sensor accuracy is determined by the

brevity of the laser pulse and the speed of the receiver. Laser rangers are, however, highly expensive when compared to ultrasonic and infrared sensors.

Range measurement devices have been widely and efficiently used to detect obstacles and perform scanning of the environment. They suit structured environments, where walls, corners and edges can be recognized for producing a map and localizing the robot. They are, however, highly susceptible to noise. Sources of errors are sensor mechanical or electrical noise, cross-talk, specular reflections and environment dynamics, all of which lead to misinterpretations. Figure 3.1-a illustrates these kinds of errors. Probabilistic sensor modeling has been adopted to overcome these errors [Thrun et al., 2005]. Figure 3.1-b presents a probabilistic sensor example that can overcome errors and uncertainties accompanied with the raw measurement of a proximity sensor.

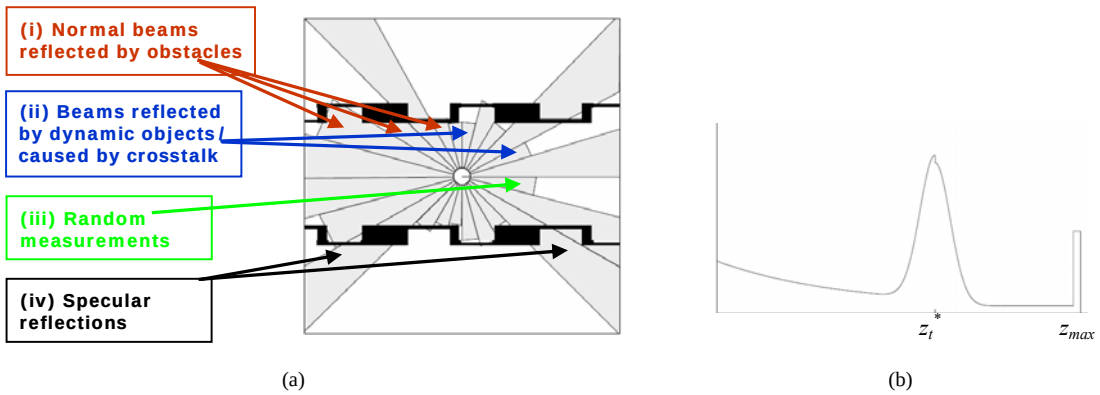


Figure 3.1. (a) Typical measurement errors of any range sensor. (b) Proximity sensor probabilistic model as a mixture distribution comprising the different errors and uncertainties associated with the range measurement. Here, z_{max} is the maximum range measurement and z_t^* is the actual beam measurement [Thrun et al., 2005].

3.2.2 Vision Sensors

Compared to range sensors as a primary sensor, vision appears to possess several appealing advantages: (i) They have virtually unlimited range and can cover large field of views at high update rates. (ii) Due to their passive nature, multiple cameras do not interfere with each other when operating in the same area. (iii) Rich information like color and texture are readily available in images. (iv) Camera systems are now available at low costs and have limited power consumption. Cameras can also provide depth information when coupled in a stereo vision head, but in this case their complexity is unacceptable when compared to range

sensors. That is why they have limited access in autonomous systems for this purpose due to the complexity in treating the vast amount of information.

Recent camera constructions exist which provide a 360 degrees horizontal field of view instead of the limited field of view offered by classical cameras. Figure 3.2 shows some examples of omnidirectional constructions that provide panoramic vision. A catadioptric sensor, such as the 0-360, offers a wide field of view usually with non-uniform spatial resolution, while a multiple camera rig can provide large fields of view with uniform spatial resolution. Approaches to panoramic photography [Brown and Lowe, 2003] stitch image shots taken separately into a single continuous image. This is computationally intensive with the usage of the RANSAC iterative algorithm to find the correspondences. Alternatively, an omnidirectional camera can be used to create panoramic art in real-time and without actuating neither the camera nor the robot. Panoramic images provide rich scene information, which can resolve many robotic navigation-related problems. They have an additional advantage of producing circular images with rotation invariant features.

The different types of imaging systems (single cameras, stereo camera, multiple camera rigs and catadioptric sensors) have been used in map building and localization because of the rich information they provide. They suit different environments and match those lacking clear structure. The main associated problem is that vision data need high processing due to the massive amount provided. Cameras do not offer data richness only, but also redundant and unnecessary data in high-dimensional spaces. It cannot be ignored that the irrelevant data, together with the high complexity of the environment (which is more revealed through vision sensing), pose difficulties in the environment processing, modeling and recognition.

3.2.3 Other Sensors

Other sensor examples include tactile sensors, force sensors, thermal sensors and active beacons. Compared to range and vision sensors, tactile, force and thermal sensors are far away from a widespread utility, although they are indispensable for the distinct human-like activities: manipulation, exploration, and response [Appin, 2007]. The importance of tactile and force sensing for manipulation appears evidently for the fine motor tasks.

Tactile sensors elicit information through physical interaction. Tactile information about materials and surface properties (e.g., hardness, thermal conductivity, roughness,

Figure 3.2. Omnidirectional vision commercial examples: (a) Ladybug consists of 6 cameras; 5 of which are radially configured on a horizontal ring and one points vertically. (b) 0-360 Panoramic Optic that provides 115 degree vertical field of view. The camera is a popular construction that consists of a perspective camera pointing to a hyperbolic mirror. (c) Stereo Omnidirectional System (SOS) with 60 color C-MOS cameras that capture raw and depth images of all directions synchronized in real-time. (d) Example of a panoramic image acquired by a common omnidirectional camera outdoors.

friction) help to identify objects. Force sensors measure components of contact forces when the fingers or arms of a gripper make contact with environment objects. These sensors are found most often at the base joints or wrist of robots, and may be distributed throughout the links of a robot. In a similar manner, forces can help to distinguish objects. Thermal sensors can be used to determine the material composition of an object, as well as to measure surface temperatures. A temperature sensor that contains a heat source can detect the heat rate absorbed by an object in room temperature where objects exhibit almost a constant temperature. This provides information about the heat capacity of the object and the thermal conductivity of the material from which it is made, and consequently makes it easy, for example, to distinguish among objects (e.g. metals from plastics).

Other than the previous group, active beacons appear to be more commonly employed sensors. These sensors are used by trilateration and triangulation navigation systems. In such navigation systems, there are usually three or more active transmitters (usually infrared) fixed at known locations in the environment and one receiver on board of the robot. Conversely, there may be one transmitter on board and the receivers fixed on the walls. Based on the time-of-flight information, trilateration systems compute the distances between the stationary transmitters and the onboard receiver. In the triangulation navigation system configuration, a rotating sensor on board of the robot registers the angles to the beacons it detects relative to the vehicle's longitudinal axis. Such navigation systems can be built simply and inexpensively. One problem with these configurations is that the active beacons need to be extremely powerful to ensure omnidirectional transmission over large distances. In addition, beacons may not be visible in many areas, a problem that is particularly grave because at least two beacons must be visible for trilateration and three for triangulation.

3.3 Vision-based Modeling Approaches

Places can be recognized in two ways, either through: (1) the objects they have which make them recognizable (e.g. a coffee machine in a kitchen), (2) holistically through a memorized pattern of the scene as a whole and not as component parts alone. Both ways describe places with a feature set, which is often referenced in the latter way by the fingerprint or the gist.

Both objects and scenes can be described through different modeling approaches that use different cues [Roth and Winter, 2008]. Earlier object modeling approaches are shape-based which try to approximate the object with a collection of three dimensional lines and planes to form geometrical primitives (e.g. boxes, spheres, cones, cylinders, surface of revolution). They are termed *model-based approaches*. An obvious limit of these approaches is the shapes they can deal with. Most real objects are complex in shape and cannot be represented in this specific manner. A second later approach tried to segment the scene into objects or homogenous parts and to characterize those segments by features (e.g. color, shape/contour, texture). The *segmentation approach* has been applied widely for several years, but the accuracy and processing of segmentation are a function of the complexity of object or scene. The object or scene may suffer variant appearance when viewed from different directions and when partially occluded by other objects.

Both the model-based and segmentation approaches are local or part-based models. A third different approach comprises a global characterization for the object or scene, and not a partial one. This is the *appearance-based* or *view-based modeling*. In this type of modeling, only the global appearance is used in which different two-dimensional information (mainly images) about the object or scene-of-interest is captured. The acquisition is sometimes complemented with the position from which the information is acquired. In this case, a direct mapping function from sensor data to pose space is preserved. For a mobile robot, the position information can be acquired through wheel encoders. Without this geometric knowledge, the only information from the environment is the images themselves, which can still be used for recognition purposes.

Opposed to part-based models, those which are based on visual appearance are significantly less expensive in computation. They bypass the exhaustive and difficult segmentation step, which might be ineffective for densely cluttered environments. They also refrain away from the non-realistic assumption of finding the ideal geometric objects. In conjunction, they provide a richer description projection for the environment rather than the limited domain provided by geometric modeling approaches. Appearance-based modeling has several other advantages. It is conceptually simpler, because it characterizes the whole scene as a template and model information of the whole image rather than searching for specific objects or parts in the image. Another significant advantage is that the global appearance characterization is a combined effect of several scene or object properties (e.g. shape, reflectance), in addition to the implicit embedding of other intrinsic parameters (e.g. pose, view point and illumination conditions). Such an issue allows easy adaptation of the environment model to the external factors through automatic learning processes.

The use of appearance based models has become recently more popular for object and scene modeling and recognition in various applications [Schiele and Crowley, 1998; Yang et al., 2004; Booij et al., 2006]. This is partly because vision sensors are becoming cheap. For mobile robot localization, it has been proved that simple appearance-based fingerprints are sufficient to recognize environment places, and hence approximately localize the robot. Appearance-based modeling can freely use local or global feature extraction methods. Those methods will be discussed in the next section.

3.4 Vision-based Feature Extraction

Feature extraction is the essential process for building any environment model. It provides a new representation of the input data for purposes of data reduction and concrete recognition.

Image features can be classified on different levels of complexity, ranging from global low-level features, such as average color intensities or histograms, via medium-level features such as edges, lines or corners, to high-level features such as objects. The more complex the representation of features is, the more efficient it is to perform in complex environments, and the more semantic it is for human understanding. Such complexity, however, has its costs in the calculation time.

The societies of signal processing and computer vision provide a wealthy set of algorithms for feature detection and description. A complete overview of the field is beyond the scope of this thesis. As a result, we outline those techniques, which deal with relevant issues in the outlined solution design (e.g. suitability for unstructured environments), and restrict them to vision data.

3.4.1 Global Features versus Local Features

Features can be extracted in two ways. They can be derived by processing the whole image or computed locally based on salient parts only. Hence, we speak of *global features* or *local features* respectively.

Global descriptors are the result of a processing for all the data in a signal. Since the whole data are included, global methods allow the reconstruction of original data from the processed features. This is the case with frequency response filters (e.g. Fourier transform, Wavelet transform), and subspace methods (e.g. Principle Component Analysis (PCA) and Independent Component Analysis (ICA)). The main idea of all these methods is to project the original data onto the frequency domain or another subspace that represents the data optimally according to a predefined criterion (minimized variance (PCA), independency of the data (ICA)). Sometimes, it is not possible to restore the data back, as in the case of statistical measures, mean values and histograms. Histograms are simple but work surprisingly well for images with distinctive colors. Many appearance-based modeling approaches rely on global descriptors and their results are striking.

Alternatively, local features are computed based on a subset of the signal data. The same global processing yields a local descriptor if it is applied partially on the data. The most common trend is to represent the appearance of the image only around a set of characteristic points known as the interest points. Interest points are identified through some saliency measure. Any type of image characterization can be used around those interest points. Among the most familiar local feature descriptors are the Scale Invariant Feature Transform (SIFT) [Lowe, 2004], the Speeded-Up Robust Features (SURF) [Bay et al., 2008], and the Gradient Location and Orientation Histogram (GLOH) [Mikolajczyk and Schmid, 2003]. They provide robust gradient-information features, relying on several scale space decompositions. The Harris corner detector [Harris and Stephens, 1988] is an efficient local detector for edge and corner extraction. It is sometimes combined with the previous descriptors for feature extraction. However, it is sensitive to the scale of the image and therefore is not suitable for building maps that can be matched from a range of robot positions.

Local features are known to be stable and observable from different viewpoints and angles. In other words, they can be robustly detected. In the recognition process, the degree of resemblance between images is a function of the number of properly matched interest points. Therefore, local features possess additional robustness to occlusions and clutter, since the non-availability of some of the interest points or the intrusion of undesired ones does not affect the recognition greatly. In a similar sense, local features are considered robust to the dynamics of operational environments, where moving people and furniture, as well as illumination, affect the sensor image. In contrast, global feature processing is known for its high sensitivity to illumination changes, besides that some extractors need extra processing if they are required to have additional invariance properties (e.g. rotation). Nevertheless, it cannot be ignored that the global descriptors are much easier in computation compared to local descriptors, which are computationally expensive in both extraction and matching.

3.4.2 Local Feature Extraction

The process of local features extraction consists of two stages: interest point detection and feature description. An interest point detector identifies characteristic points in the image that should be capable of being re-detected, in spite of various transformations (e.g. rotation and scaling) and variations in illumination. The role of the descriptor is to provide characterization from the local patches located at the detected points.

The following subsections introduce the family of scale-invariant feature extraction methods. They depend on the scale-space theory, which provides a framework for analyzing signals at multiple scales. Since no a priori information about the scale is usually available, it is necessary to create a multi-scale representation on the basis of the original image, in which the fine-scale structures are successively suppressed with the increase of scale. All the members of this family show high robustness against image rotations, illumination changes and perspective deformations.

The main idea of those techniques is based on convolving the image with a Gaussian kernel of increasing variance in order to smooth the image. This blurring Gaussian operator simulates the idea as if a viewer is moving away from the scene. Convolving with Gaussian filters is a low-pass filtering. Therefore, it is ideal for suppressing the high frequency components in real noise that may occur in the imaging process. It additionally makes the Gaussian derivatives in the scale space more stable.

In what follows, the most applied local feature extractors are explained. The two construction phases, the interest point localization and the feature descriptor computation are described for each technique.

3.4.2.1 Harris-Laplace Interest Point Detector

Proposed back in 1988, the Harris corner detector [Harris and Stephens, 1988] has shown good performance in extracting corners and junctions. In [Ballesta et al., 2007], it is suggested as the most suitable interest point detector for visual SLAM. The detector is based on the second-moment matrix (also called the auto-correlation matrix), and can be used arbitrarily in conjunction with any feature description. The detector exhibits, however, sensitivity to variations in the image resolution (i.e. scale).

[Mikolajczyk and Schmid, 2004] combined the reliable Harris detector with automatic scale selection [Lindberg, 1998] to provide a scale-invariant interest point detector, which they coined Harris-Laplace. The scale-adapted Harris measure is a function of the second moment matrix:

$$\mu(x,y,\sigma_I,\sigma_D) = \sigma_D{}^2 g(x,y,\sigma_I) * \begin{bmatrix} L_x^2(x,y,\sigma_D) & L_x L_y(x,y,\sigma_D) \\ L_x L_y(x,y,\sigma_D) & L_y^2(x,y,\sigma_D) \end{bmatrix} \tag{3.1}$$

where *g* is a Gaussian kernel of scale σ_I (integration scale), $L(x,y,\sigma_D)$ is the smoothed image computed using a second Gaussian kernel with scale σ_D (differentiation scale), the operator * denotes convolution, and L_x and L_y are the respective derivatives of the smoothed image (i.e. gradient) in the *x* and *y* directions. The second moment matrix is the matrix with off-diagonal entries equal to the product of L_x and L_y, and diagonal entries equal to the squares of the respective derivatives.

A function *R* for the cornerness at a given point and scale is defined based on the determinant and the trace[2]:

$$R = \det(\mu(x,y,\sigma_I,\sigma_D)) - \alpha \,\mathrm{trace}^2(\mu(x,y,\sigma_I,\sigma_D)) \tag{3.2}$$

where α is a constant. Corresponding interest points are located by finding the local maxima of *R* at a given characteristic scale. This characteristic scale is selected by searching over multiple scales σ_I^n for a local extremum of Laplacian-of-Gaussian (LoG) responses:

$$\det(LoG(x,y,\sigma_I^n)) = \sigma_I^{n\,2} \det(L_{xx}(x,y,\sigma_I^n) + L_{yy}(x,y,\sigma_I^n)) \tag{3.3}$$

where L_{xx} and L_{yy} are the second order derivatives in their respective directions.

3.4.2.2 Scale Invariant Feature Transform (SIFT)

Scale Invariant Feature Transform (SIFT) [Lowe, 2004] is a popular local feature detector and descriptor in the computer vision society. SIFT features correspond to highly distinguishable image locations that are extracted efficiently and with great stability across wide variations of viewpoint and scale. Those locations are detected by searching for peaks in an image $D(x,y;\sigma)$, which is obtained by subtracting two neighboring smoothed images in the scale space, and which is separated by a constant multiplicative factor *k*, such that at a particular σ :

$$D(x,y;\sigma) \;=\; L(x,y;k\sigma) \;-\; L(x,y;\sigma) \tag{3.4}$$

where σ denotes the scale blurring parameter. The image scale space *L* is produced by the convolution of the gray-scale image $I(x,y)$ with a Gaussian kernel *G* of the varying σ to produce a smoothed image version:

[2] The trace of a square matrix is the sum of the elements on the main diagonal.

$$L(x, y; \sigma) = G(x,y; \sigma) * I(x,y) \tag{3.5}$$

Such that:

$$L(x, y; 0) = I(x,y) \tag{3.6}$$

$$G(x,y; k\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{3.7}$$

From the current scale space, a new scale space is constructed using the Difference-of-Gaussian (DoG) function[3] *D*:

$$D(x,y;\sigma) = (G(x,y;k\sigma) - G(x,y;\sigma)) * I(x,y) \tag{3.8}$$

The procedure is repeated on multiple octaves, with each following octave having an image with half the resolution of the image in the current octave.

The candidate feature locations - called *keypoints* in the SIFT framework - are obtained by searching for local maxima or minima in $D(x,y,\sigma)$ for each pixel neighborhood at a given scale, plus the neighborhood of the two adjacent scales – i.e. eight in the current image, and nine in each of the scales below and above (see figure 3.3). That is to say, the simultaneous selection of interest points $(\hat{x}, \hat{y})$ and scales $\hat{\sigma}$ is performed according to the operator:

$$(\hat{x}, \hat{y}, \hat{\sigma}) = \arg\min\max local_{(x,y,\sigma)} (D(x, y, \sigma)) \tag{3.9}$$

In the second stage, nearby data for candidate keypoints are interpolated to determine their position accurately. Points with low contrast and poor localization are discarded. Finally, each keypoint is assigned an orientation, such that the descriptor is represented relative to this orientation, and achieves invariance to rotation. Orientation is calculated from a histogram of local gradients from the closest smoothed image $L(x,y,\sigma)$. For each image sample $L(x,y)$ at this scale, gradient magnitude $M(x,y)$ and orientation $\theta(x,y)$ are obtained using pixel differences:

$$M(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \tag{3.10}$$

$$\theta(x, y) = \tan^{-1} \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \tag{3.11}$$

[3] The DoG is a close approximation of the LoG producing similar response effects. LoG is computationally expensive, since smoothed images should be computed anyway. In DoG, the calculation is reduced to image subtraction which significantly accelerates the computation process [Lowe, 1999].

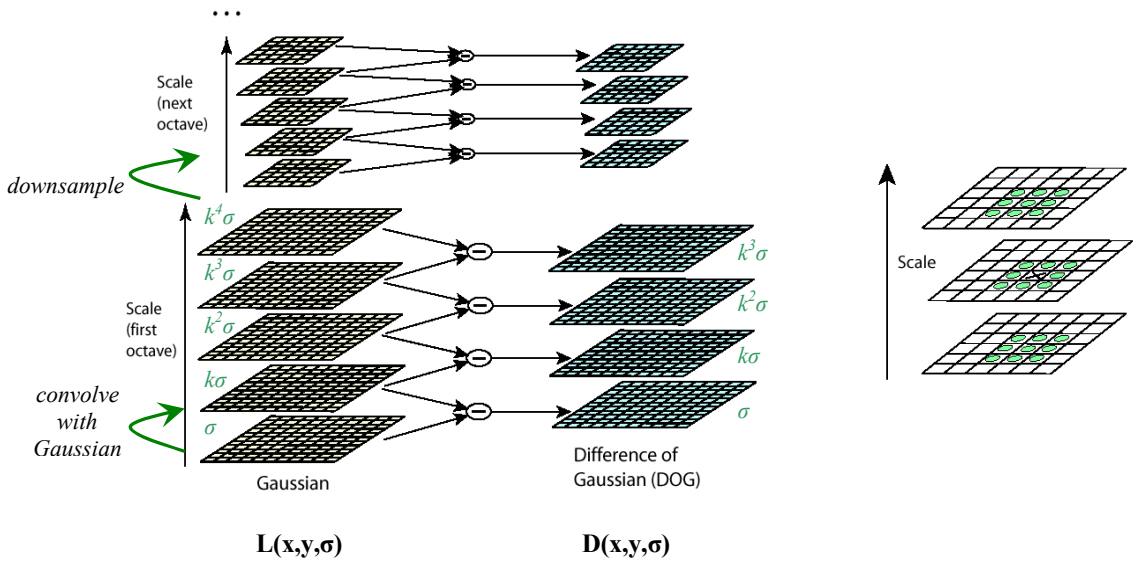


Figure 3.3. SIFT Processing. Left: For each octave of scale space, initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the DoG images on the right. After each octave, the Gaussian image is down-sampled by a factor of 2, and the process is repeated. Right: Maxima and minima of the DoG images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles).

Finally, the descriptor is constructed by computing local orientation histograms (8-bin resolution) for each element of a 4x4 grid overlaying a 16x16 neighborhood of the keypoint. This 4x4 window operation in the local neighborhood yields a 128-dimensional feature vector. The features are normalized to unit length in order to reduce the sensitivity to image contrast and brightness changes in matching stages. Figure 3.4 illustrates the descriptor generation procedure of keypoint for a 2x2 window. Figures 3.5 and 3.6 show an image decomposition example and the corresponding generated keypoints by the algorithm.

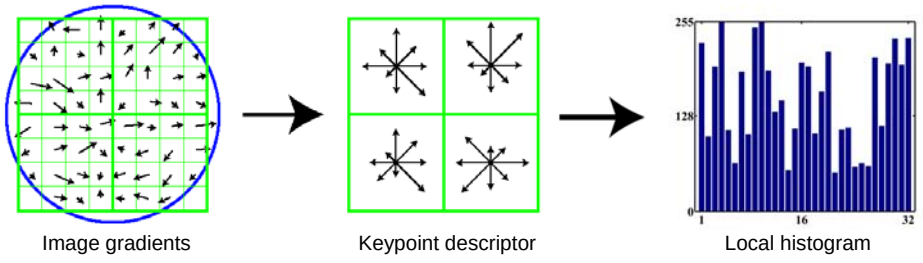


Figure 3.4. SIFT feature descriptor generation

A main drawback of SIFT, as many other local descriptors, is the high computational cost, besides the huge number of features that densely cover the image over the full range of scales and locations. Features' size usually influences time performance to a great extent when matching the features in recognition and retrieval systems. A typical image of size 500x500 pixels will give rise to about 2000 stable features according to the author [Lowe, 2003].

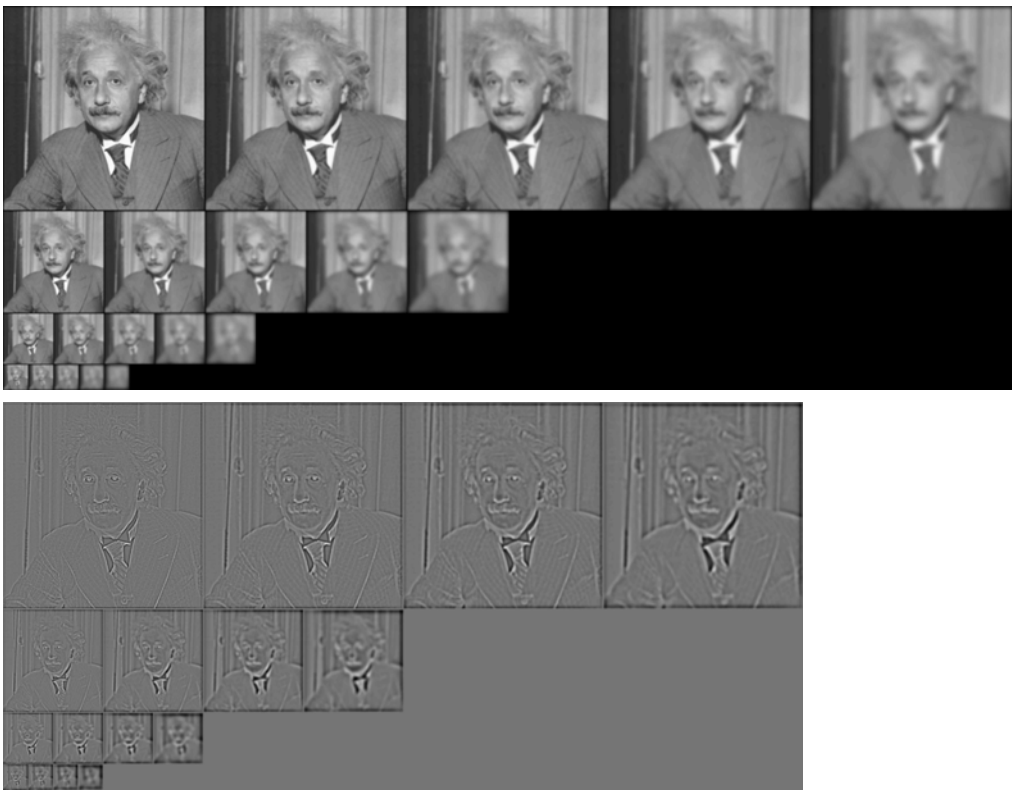


Figure 3.5. The scale space constructed by combined smoothing and sub-sampling and the corresponding DoG levels obtained by subtracting neighborhood images.

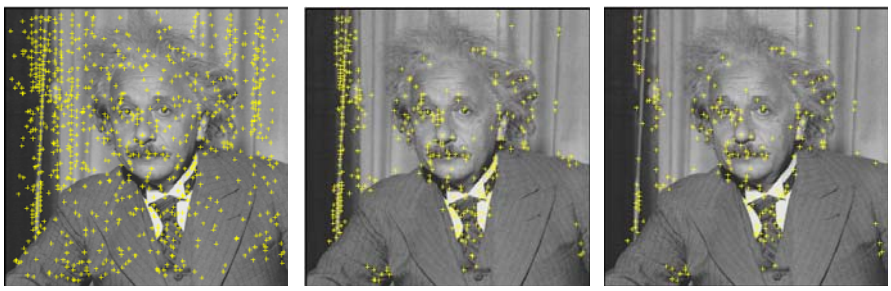


Figure 3.6. SIFT feature extraction example. (a) Peaks of DoG across scales. (b) Remaining keypoints after discarding low contrast points. (c) Remaining keypoints after discarding edge responses.

Therefore, reducing the quantity of generated keypoints without negatively affecting the general accuracy performance of recognition is highly desired for time critical systems. This problem is one of the main concerns in this thesis.

3.4.2.3 Speeded Up Robust Features (SURF)

Speeded Up Robust Features (SURF) is a local feature detector and descriptor recently introduced in [Bay et al., 2008]. It utilizes the same basic concepts as SIFT. It uses, however, several approximations and shortcuts to shorten the computation time. The detector is based on the determinant of the Hessian for selecting the location and scale of the interest points. For a given point *(x, y)* in an image *I*, the Hessian matrix[4] H_L at scale σ is defined through the second order derivatives as follows:

$$H_L(x,y,\sigma) = \begin{bmatrix} L_{xx}(x,y,\sigma) & L_{yx}(x,y,\sigma) \\ L_{xy}(x,y,\sigma) & L_{yy}(x,y,\sigma) \end{bmatrix} \tag{3.12}$$

where $L(x,y,\sigma)$ is the convolution of the Gaussian $G(x,y,\sigma)$ with the image *I* for the point (x,y) at scale σ, defined by equations (3.5) and (3.7).

In contrast to SIFT, which approximates LoG with DoG for the interest point detection, SURF approximates the second order Gaussian derivatives of the Hessian matrix with box filters. An example of one of 9x9 box filters that represent the lowest scale (i.e. highest spatial resolution) is shown in figure 3.7. Image convolutions with these box filters can be computed rapidly using integral images [Messom and Barczak, 2008].

The scale space ($\det(H_L)$) is analyzed by up-scaling the filter size, rather than iteratively reducing the image size. Precisely, the filter size is doubled for each new octave generated from the lowest scale up to the highest. Because of employing box filters and integral images, the same filter does not have to be iteratively applied to the output of a previously filtered layer. This direct processing provides faster execution than SIFT, which requires the computations of the images $L(x,y,\sigma)$ for the different scales and octaves in order to find the extreme points in the DoG image.

An integral image $I_{\sum}(x,y)$ at point (x,y) contains the sum of the pixels of the input image $I(x,y)$ above and to the left of *x* and *y*, inclusive, defined by:

[4] The Hessian matrix is the square matrix of second-order partial derivatives of a function. In this context, it represents the convolution of the Gaussian second order derivatives with the image I at a given point and scale.

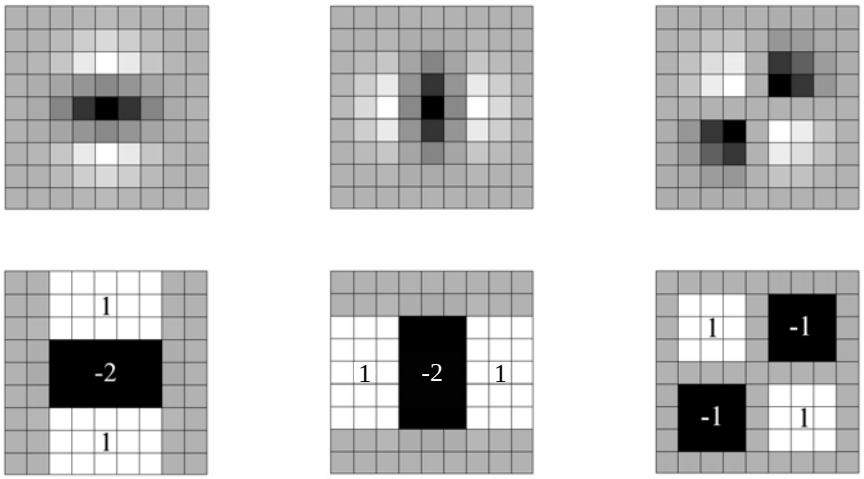


Figure 3.7. SURF Processing. Top: Gaussian second derivatives in horizontal x, vertical y and diagonal xy directions respectively. Bottom: Their corresponding box filters. Back refers to negative values, white to positive values.

$$I_{\Sigma}(x,y) = \sum_{i=1}^{i \le x} \sum_{j=1}^{j \le y} I(i,j) \tag{3.13}$$

Box filters of any size can be fast computed by using the integral image (the sum of a block in $I(x,y)$ with the top-left corner at (x_1,y_1) and the bottom-right corner at (x_2,y_2)):

$$I_{\Sigma}(x_1-1, y_1-1) + I_{\Sigma}(x_2, y_2) - I_{\Sigma}(x_1-1, y_2) - I_{\Sigma}(x_2, y_1-1) \tag{3.14}$$

Using the integral image, SURF approximates the different levels of scale space by adjusting the size of the box filters. Once all images of the scale space have been computed, keypoints (local maxima) are localized by a non-maximum suppression in a 3x3x3 neighborhood around each sample point.

$$(\hat{x}, \hat{y}, \hat{\sigma}) = \arg\min\max local_{(x,y,\sigma)} \left(\det(H)\ L(x,y,\sigma)\right) \tag{3.15}$$

Finally, the local maxima found in the approximated Hessian matrix determinant are interpolated in scale and image space (i.e. localization of keypoints is improved using the second-order Taylor expansion as in SIFT).

For the descriptor, SURF uses Haar-wavelet responses to determine the keypoint orientation. The Haar wavelets can be fast computed via integral images, similar to the

Gaussian second order approximated boxes. Two-dimensional Discrete Wavelet Transform is applied in a 4x4 quantized sub-regions of a local neighborhood, and the responses in only two directions (horizontal direction d_x and vertical direction d_y) are preserved after they have been weighed by a Gaussian. Each sub-region has a four-dimensional descriptor vector consisting of the sum and the absolute values of responses over each sub-region:

$$v = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|\right) \tag{3.16}$$

Stacking the vectors for all sub-regions gives a descriptor for the interest point of length 64 (the standard SURF-64). The final descriptor is obtained by normalization to unit length.

3.4.3 Feature Evaluation and Selection

Systems that navigate using high-resolution sensors have to cope with a massive data flow produced. To reduce the data volume somehow, low-level (e.g. edges, planes) and high-level (e.g. doors, tables) features are predefined and included in advance in the environment model. Later on, the sensor processing system has to be able to identify those features in the image data it perceives.

Nevertheless, the navigation system using the designed feature space may still perform inefficiently. The feature space may contain redundant or misleading information, and can still be reduced somehow. Explicit definition of features or the existence of a semantic representation is not fulfilled in every modeling approach, especially when the features are defined on a low level of abstraction (e.g. point features or dense scans). This poses more difficulty to understand the data. Intelligent data mining techniques are designed to solve these issues by extracting much more meaningful information [Han and Kamber, 2006]. They design systems with better performances, which comply with the intelligent human decisive behaviors.

In the field of robotics, many studies apply feature extraction and representation methods, inspired by the large recognition domain of objects and scenes. Few do pay attention to the combined quality and size of extracted features for the target application. No efforts have been exerted in applying feature selection criteria that contribute to more efficient environment modeling. The studies assume that a domain of features is qualitative enough for employment, without paying attention to its size or its information content. The

literature presents dozens of robotic applications' examples, which all use high-dimensional local descriptors directly (i.e. the way they are extracted) for topological, metric and hybrid representations [Ramisa et al., 2008; Murillo et al., 2007a; Andreasson and Duckett, 2004; Se et al., 2001; Ballesta et al., 2007; Goncalves et al., 2005; Wang et al., 2006; Valgren, 2007; Werner, 2010; Zivkovic et al., 2005].

Feature selection – also identified as a variable selection and feature reduction – is a group of machine learning techniques for selecting a subset of relevant features to build robust learning models. The process is based on including some evaluation procedure or criteria. Feature selection algorithms typically fall into two categories: *feature ranking* and *subset selection* [Guyon and Elisseeff, 2003]. The first category ranks the features according to an assigned metric and then eliminates features that do not achieve an adequate score. The latter performs an exhaustive search for the set of possible features to find the optimal subset. Regarding time complexity, metrics can be considered more efficient. Popular filter metric examples are correlation, entropy and mutual information.

From the general perspective, the objective of variable selection is three-fold: (i) improving the prediction performance of predictors, (ii) providing faster and more cost-effective predictors, and (iii) providing a better understanding of the underlying process that generated the data. The most active areas applying variable selection are gene selection from microarray data and text categorization. Yet its application is needed in other applications as well, where rich data are available like vision and highly complex feature vectors, and where perceptual aliasing contributes to higher uncertainties or misrecognitions.

The research work regarding dimensionality reduction for local descriptors and features pruning is relatively limited. In [Ke and Sukthankar, 2004], PCA has been applied to reduce the SIFT dimensionality for the purpose of individual object detection. The study recommended the use of 20 dimensions instead of 128 to the specified application. For the same purpose, [Ayers and Boutell, 2007] used Linear Discriminant Analysis (LDA) for indoor scene classification, and experimented with feature pruning using a clustering technique. The approach, however, was reported as unsuccessful. In [Ledwich and Williams, 2004], the rotational invariance step of the SIFT algorithm is eliminated resulting only in small reduction in computation complexity of the algorithm but not in the number of features.

Some work presents trials to reduce the number of computations, but not based on a quality measure. In [Bennewitz et al., 2006], SIFT features are down-sampled through random sampling from every cell in a feature-based grid map. A drawn feature is rejected if there is already a similar feature within the grid cell. A maximum of 20 features for each cell are sampled. The idea seems logical to reduce the features, but obviously not qualitative. The eliminated feature might have been distinguishable for cell disambiguation. A second trial for pruning the place images themselves to reduce the map size is presented in [Booij et al., 2006]. In [Sala et al., 2004], the minimal number of landmarks in environment regions is preserved such that every image captured in that region can still be localized. In other words, features that persistently appear in all images enclosing a region are regarded as good features, and hence selected.

[Werner, 2010] claimed, on the one hand, that incorporating neighborhood relations can solve perceptual aliasing and disambiguate places. It is true, but this is not related to the capability of data to resolve ambiguities. On the other hand, the work which introduces classifiers to classify the different places can be counted as methods that generate models based on quality measures [Pronobis et al., 2006].

In the same sense, few significant works regarded data compression to account for the undesired computational complexity of local or huge size features. A Bag of Words model (BoW) inspired by text retrieval systems has been applied to enhance localization using vision-based local features in [Filliat, 2007]. Figure 3.8 shows an example of BoW modeling, where an image can be described through a set of compressed codewords by involving clustering techniques.

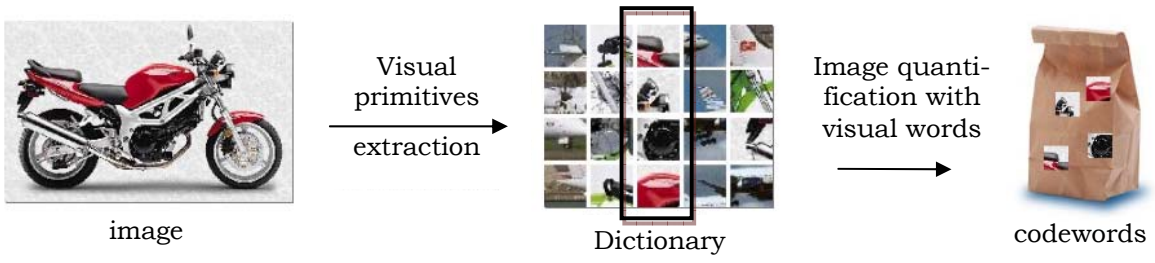


Figure 3.8. Bag-of-visual-words approach. In the first step features are computed from images. The features are clustered into visual words in the second step. Finally the image representation can be computed as a histogram over visual words. (Source: http://cogrob.ensta.fr/index.html.)

From the previous related work in the area of data selection and reduction, it is clear that comparable approaches in the robotic domain, which concerns the combined size and quality of features for target applications, are not widely implemented so far.

3.5 Information Theory

Information theory is concerned with the study of information production and transmission. It is regarded as a general framework since it is based on the statistical properties of the system and not its explicit description. The development of the standard communication model by Shannon did not only establish theoretical bounds on the limits of data compression and transmission over a noisy channel, but also contributed to several applications in other communities concerned with the quantification of information [Verdu, 1998]. The application of this model can be quite useful in judgments involving the economy of information processing systems.

Figure 3.9 depicts Shannon's standard communication system, sometimes called *source-channel model*, for a noisy memoryless channel [Shannon, 1948]. The input to the channel is a space of messages to be transmitted normally after being encoded. The output is the space of messages received over the channel after being decoded in a specific time. The transmitter and receiver sides should agree on a certain alphabet (i.e. symbol set) that forms the ingredients of the message content. In the communication system shown, the source selects a desired message from the set of possible messages which the transmitter changes into a signal that is actually sent over the communication channel. The receiver changes this signal back into a message, and hands it to the destination. Ordinarily, the signal is perturbed by noise during the transmission. Consequently, the received signal and the received message are not necessarily the same as those sent out by the transmitter. This results in a possible misinterpretation at the receiver's side.

The source is memoryless in the sense that the occurrence probability of a symbol does not depend on previous symbols. In other words, the occurrence of alphabet is independent and identically distributed (i.i.d). From the stochastic perspective, the channel's behavior can be characterized by a conditional probability distribution $p(Y|X)$, where X is a random variable representing the channel input and Y is a random variable representing the channel output. What the information theory seeks is trying to maximize the transmitted information

rate over the channel. It is something which has to do with the encoding of information, such that the channel is optimally utilized in terms of a maximum transmission with minimal error. The information theory achieves that by matching the source to the channel in a statistical way.

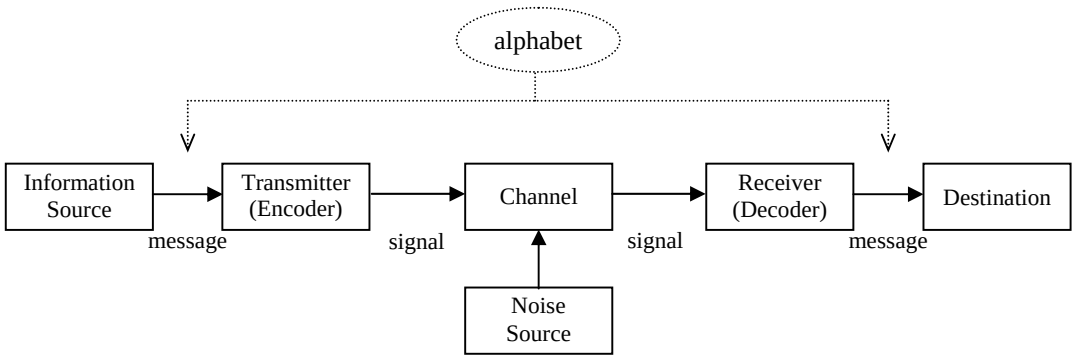


Figure 3.9. Schematic diagram for a general communication system.

Several elements are defined in the framework of the information theory; information content, entropy, conditional entropy and mutual information. These elements are defined for a discrete channel as follows:

For the possible states of the random variable X, the information content per arbitrary symbol x_i is defined as:

$$i(x_i) = -\log_2 p(x_i) \tag{3.17}$$

The information in this sense represents the outcome of a selection among the finite number of possibilities of the variable X [Verdu, 1998]. A similar relation can be given for the information content per output symbol y_j:

$$i(y_j) = -\log_2 p(y_j) \tag{3.18}$$

The entropy is defined as the average information content in the random variable and is sometimes called uncertainty:

$$H(X) = -E(\log_2 p(x)) \tag{3.19}$$

$$= -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \tag{3.20}$$

where $0 \le p(x_i) < 1$ and $\sum_{i=1}^{n} p(x_i) = 1$, and n is the size of the distribution of X. Similarly, the entropy can be defined for the output variable Y of distribution size m.

$$H(Y) = -E(\log_2 p(y)) \tag{3.21}$$

$$= -\sum_{j=1}^{m} p(y_j) \log_2 p(y_j) \tag{3.22}$$

Entropy is also regarded as a measure of the variation, dispersion, or diversity of a probability distribution of observed events. Normally, it is applicable to measure different quantities, including randomness, uncertainty, ignorance, surprise or information.

The conditional entropies *H(X|Y)* and *H(Y|X)* are called the Equivocation and Dissipation simultaneously, and are defined by:

$$H(X \mid Y) = -\sum_{i=1}^{n} \sum_{j=1}^{m} p(x_i, y_j) \log_2 p(x_i \mid y_j) \tag{3.23}$$

$$H(Y \mid X) = -\sum_{i=1}^{n} \sum_{j=1}^{m} p(x_i, y_j) \log_2 p(y_j \mid x_i) \tag{3.24}$$

The mutual information, sometimes called the transmission or transinformation, is a measure for the information processed by the channel:

$$I(X,Y) = \sum_{i=1}^{n} \sum_{j=1}^{m} p(x_i, y_j) \log_2 \frac{p(x_i, y_j)}{p(x_i) p(y_j)} \tag{3.25}$$

It is also a measure of the dependency between two random variables. Hence, it is a quantification measure for the degree of structure.

Some relationships relate the different information-theoretic elements to each other. These relationships are illustrated in figure 3.10, and are summarized by the following.

$$H(X,Y) = H(X) + H(Y \mid X) = H(Y) + H(X \mid Y) \tag{3.26}$$

$$I(X,Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X) \tag{3.27}$$

$$H(X) \geq H(X \mid Y) \tag{3.28}$$

$$H(Y) \geq H(Y \mid X) \tag{3.29}$$

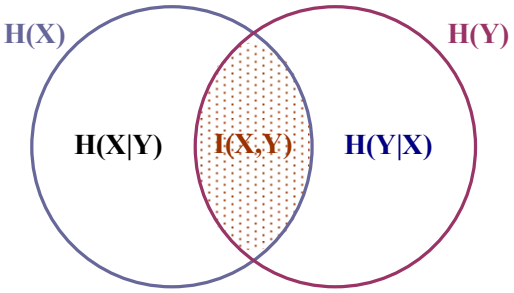


Figure 3.10. Entropy, conditional entropy and transmission relationships.

Some special cases of a channel exist. A *noiseless* channel has coinciding input and output sets resulting in:

$$H(Y \mid X) = H(X \mid Y) = 0 \tag{3.30}$$

and

$$H(X,Y) = H(X) = H(Y) = I(X,Y) \tag{3.31}$$

A channel is called *lossless* if $H(X|Y) = 0$; and *deterministic* if $H(Y|X) = 0$. A noiseless channel is, therefore, both lossless and deterministic. The real situation is that the channel is stochastic. For a strongly noise-corrupted or useless channel, the input and output sets are disjoint or mutually independent. i.e. $I(X,Y) = 0$.

And hence,

$$H(X,Y) = H(X) + H(Y) \tag{3.32}$$

$$H(X \mid Y) = H(X) \tag{3.33}$$

$$H(Y \mid X) = H(Y) \tag{3.34}$$

The channel capacity is the maximum *transmission rate* taken over symbol probabilities, if the average symbol duration is taken as unity.

$$C = \max_{p(x_i)} I(X,Y) \tag{3.35}$$

Other related quantities are the channel redundancy:

$$R_C = C - I(X,Y) \tag{3.36}$$

and the channel efficiency:

$$\eta_C = \frac{I(X,Y)}{C} \tag{3.37}$$

Information is highly important in the study of complex systems and structures. The randomness and unpredictability of a system (i.e. entropy) do not completely capture the correlational structure in its behavior, however conditional entropy and transmission do. Information theory has been applied in robotics to choose the best actions in active sensing [Seekircher et al., 2011; Denzler and Brown, 2002; Davison, 2005], speed up exploration [Sujan and Dubowsky, 2005], and to cooperatively share the most useful information between communicating robots for map building [Rocha et al., 2005]. Information distances, such as the Kullback-Leibler distance, have been sometimes used for the general comparison between distributions. An interesting application of information distances is used in [Lungarella and Pfeifer, 2001; Olsson et al., 2004], where structure of the environment (e.g. vertical contours) is detected through distance measurements in the vision sensor plane.

From the control engineering point of view, it is useless to deliver information in a system at a rate much higher than what is actually required by the task. Therefore, much effort should be dedicated to filtering the input data and extracting the relevant part out of it for avoiding unnecessary actions [Badreddin, 2002]. In the proposed work, the problems of information selection and filtering are viewed from the source-channel perspective which is combined with machine learning, in order to select the robust information that maximizes the channel transmission. More specifically, what is sought is the maximization of transmission between the system states which represent the channel source from one side and the environment observation domain which represents the receiver from the other side. A quality-based and less complex environment model should be generated from such information-theoretic environment modeling perspective.

3.6 Summary

This chapter has introduced some preliminaries for the proposed work. The topics discussed are environment modeling basics, exteroceptive sensors, visual modeling approaches, feature extraction, and information theory. In basic terms, the chapter lays out the idea that environment modeling can be robustly and efficiently established based on concepts of learning, general location appearance, and proper data selection independent of its modality. The late issue, regarding data selection, has been hardly tackled in the field of robotics on a fair scale as reported by the literature. A proposal for the involvement of information-theoretic concepts is suggested. It will add optimal refinements for the data and information engaged, in a way that should increase the efficiency of environment modeling and localization in robotics.

Chapter 4

Information-Theoretic Approach for Environment Modeling and Localization

This chapter tackles the navigation-related questions: What is the suitable representation for a robot's operating environment, and what is the global position of robot in the environment? The answers to the questions are presented in the frame of an information-theoretic environment modeling solution for localization purposes. The proposed modeling solution targets constructing information-rich and compact environment models and map projections, which manage to achieve combined accuracy and computational efficiency localization performances. The publications of [Rady et al., 2008; 2009; Rady and Badreddin, 2010] summarize the solution approach and included methods presented in this chapter.

4.1 Introduction: 'Why Information Theory?'

Generally, robots work in various complex environments characterized by lots of details, unstructuredness and dynamics. The sensors used by the robot for perception often provide large and noisy data streams depending on the corresponding environment detail. The complexity of the environment, together with the uncertainty of perception, represents a

challenge for navigation. For this purpose, a computationally efficient method is required for building robust models that can be employed in real-time.

Most robot navigation methods are model-based. This means an explicit representation about the space in which navigation is to take place should be defined. An information-rich environment model is definitely of great significance for the related navigational tasks including self-localization and path planning. Information-rich models do not imply that they just contain massive or heterogeneous data. On the contrary, excessive data increase uncertainties associated with the system states and degrade the computational performance. Richness of information, in this sense, is defined by the right data in an exact amount that resolve states' ambiguities and accomplish the task successfully. Accordingly, a model constructed on this basis is expected to issue better navigational capabilities in terms of both accuracy and computational performance.

The previous argument is crucial, but unfortunately hardly tackled in the robotic real-world applications on a fair scale. In [Siciliano and Khatib, 2008], three explicit challenges are mentioned as characteristics to be fulfilled by environment models. The models must (i) be compact so that they can be used efficiently by other components in the system; (ii) be adapted to the task and to the type of environment *(illustrative example: it is irrelevant to model the environment as a set of planes for a robot operating in a natural terrain)*; (iii) accommodate the inherent uncertainty of both sensor data and robot states. The three requirements can be viewed as equally-important axioms for building environment models of efficient use. Similarly, in [Badreddin, 2002], the need for methods accommodating the right amount of information to fit exactly that required by the task, *and not higher*, has been highlighted. Despite that, the second requirement of the prior, which coincides with that of the latter, did not gain the relevant attention.

This work targets such an unattended subject in the environment model building and uses it in localization purposes. It aims at *'constructing compact, less-complex and information-rich environment models for enhancing localization performance'*. The desired objective is met by proposing an environment modeling solution based on the information theory. The theory supplies measures that can analyze the perceived environment data in relation to the given task in quantitative terms. Based on the analysis, the quality and quantity of the information units to be perceived are identified and controlled. Hence, consecutive

data selection, filtering and possibly compression assist constructing the desired model that will contain the right and minimal amount of data fitting to the task. This implies adequate accuracy and computational complexity performances. The powerful advantage about information theory is that it lays a general infrastructure for the problem. It does not put constraints about following a specific modeling approach for the environment characterization or a specific type of sensor to be used. The theory focuses on extracting the meaningful and effective information units in a statistical point of view only, independently of any sensor types or feature extraction methodologies.

The idea behind employing an information-theoretic solution for environment modeling is driven by the intrinsic human notion that distinguishing information is related to its frequency of occurrence. A single occurrence of a special landmark in a certain place, for example, augments a unique description and identification of this place. In a similar manner, the existence of some objects in a place and their absence in another creates a way for place differentiation, even irrespective of the number of objects' occurrences in the place. It is often too that some negative information can provide this place identification (e.g. an object presence in several places except for one distinguishes this late place). High-level real world examples are a coffee machine that is most probably found in a kitchen indoors, and platform number panels and clocks that are found on train platforms in railway stations outdoors. The information is simple, yet induces high localization accuracy at the same time. Thus, the general idea is to capture only this distinguishing data and employ them in an environment model that can resolve the ambiguities in the states (e.g. robot location, identity of an object/a feature) in few computing steps. The idea can be realized through information and communication concepts of the information theory, where helping metrics exist. Those metrics can indicate informativeness of data as the mutual relationship between the perceived data and the task.

From another perspective, an information-theoretic-based environment modeling provides a concurrent solution for reducing high-dimensional feature spaces through pruning. Data are frequently dense, especially if the robot possesses several sensory devices. Using information-theoretic metrics, it is possible to extract the most relevant data part solely and discard the other which contributes to extra computational overhead.

In the general view, fitting the information content to the task and data filtering via information-theoretic measures affords two significant gains: (1) Minimization of uncertainties arising from certain problems, such as perceptual aliasing and the corresponding data association problem. Minimizing uncertainties directly implies increasing the task accuracy. (2) Minimization of time and space complexities involved, which means increasing the computational efficiency of task execution. Those two aspects have been identified as persisting challenges that confront model building and localization in figure 1.6.

Fitting the information content to the task and data filtering, through the proposed information-theoretic solution approach, also share one common objective. They try to maximize the accuracy of a task with the minimal use of computational resources. In counterpart to maximizing the task's accuracy, the accuracy can be set to meet a specified or minimum level in accordance with the resources available at hand. This means performance parameters (accuracy, complexity) can be simultaneously tuned with the need (e.g. higher accuracy for a large afforded memory space, a smaller memory at a lower accuracy cost, picking up a sensor that maximizes accuracy, or one that minimizes power consumption, etc). In other words, information rates inside systems can be adjusted, such that they fit exactly to the amount required by the task or to the available resources.

The explained expectations and advantages of involving an information-theoretic solution initiate primary motivation to engage the theory in environment model building. This chapter introduces the information-theoretic approach with a structured solution proposal, which is applied on the topological level of model building and localization. Nevertheless, the approach is still extensible for metric model building and metric localization, as will be introduced in a late chapter.

The proposed topological modeling approach includes basic structural filtering components for feature evaluation and feature compression, in addition to the common feature extraction component. A generated topological map for localization will finally contain a reduced selected feature set. The feature set is the output of information-theoretic evaluation component that accommodates supervised learning and an entropy-based criterion for filtering. The output of this component is called *entropy-based features*. The evaluation is, furthermore, implemented in a way that allows features to preserve a second compressed format, which is termed *codewords*. Those codewords enable faster localization with

maintained accuracy. Hence, the presented solution not only generates quality-based feature maps, but also compact ones. Localization is conducted through topological node matching, where the robot compares sensed features to the stored map in order to identify its location. The robustness and accuracy of localization using the two feature formats are discussed equally in this chapter, and the computational savings reflecting space and time complexities are highlighted. A single vision sensor and a robust local feature extraction methodology are suggested to demonstrate the approach. This choice provides collateral advantage of modeling unstructured environments using their natural features without external aids.

This chapter is structured as follows: First, a view of how the information theory is applied for environment modeling is presented in section 2. Next, some design aspects of the solution are outlined in section 3. The problem statement is mentioned and mathematically formulated, together with a description for the solution approach, in section 4. Based on the given aspects and problem-solution description, a general solution structure is laid out in section 5, with the details of its structural components described in section 6. Section 7 states the performance metrics employed for measuring the accuracy of localization. Section 8 describes the test environment, data acquisition and the reference method for comparing the presented approach, whereas section 9 presents the experimentations and results. Finally, section 10 summarizes the chapter.

4.2 Application of Information Theory to Environment Modeling

Location information is affected by two types of uncertainty which includes ambiguity and imprecision [Pèrez, 2005]. Statistical variations are the main source of ambiguity [Warren, 2007]. For the localization problem, it is claimed that ambiguity is inherently related to the environment structure (e.g. several locations contain identical objects or are similar in the general appearance – see example in figure 1.4). Definitely, resolving ambiguity is a difficult problem. Imprecision, on the other hand, arises from approximations of the used techniques and algorithms, parameterization, as well as noisy measurement data.

An environment is modeled based on an information-theoretic formulation by regarding the location identification process as a communication channel that attempts to minimize the location ambiguity. The communication channel will quantify the mutual relationship between its input and output state variables. The quantification will assist minimizing

ambiguity in the input state variables through proper filtering of the output. In analogy to Shannon's classical communication channel which maximizes the information sent (see figure 3.9), the objective of the constructed channel is to maximize information rate between an input location space and a corresponding output feature space. Consequently, the channel will maintain minimized data rates for the map building, while keeping information rates above minimum for the online localization process.

Figure 4.1 introduces an exemplar noisy channel for localization purpose, and in other terms for recognition purpose. Following Shannon's channel definition, the input is a set of possible location instances (robot possible positions), and the outputs are a set of corresponding environment observation instances. The inputs and outputs can be either discrete state or continuous state that undergoes discretization for further processing. The example shows a discrete input and continuous output channel. Let's assume that the input random variable X represents the location with the sample space $\{x_1, x_2, \ldots, x_n\}$, and the output random variable Z represents the observation, with the sample space $\{z_1, z_2, \ldots, z_m\}$. Each single input event $x_i' \in X$ will be 'transformed' by the channel to the corresponding possible output set $z_j' \subset Z$; $1 \leq i \leq n$ and $1 \leq j \leq m$, whereas a joint or conditional probability that represents the channel will encode the characteristics of this transmission. In reality, the outlined channel represents a coding channel, since the location transmission over the channel is a kind of signature formation or feature extraction that processes the given input locations to produce observations.

For the outlined channel, we define the following unconditioned and conditioned entropies. $H(X)$ is the entropy of the location random variable and is an indication of the uncertainty as well as the average information per location. Similarly, $H(Z)$ is the entropy of the observation random variable and is an indication of the uncertainty and the average information per observation. $H(X,Z)$ is the joint entropy between the location and observation variables and is the indication of the overall uncertainty of localization or recognition. This joint entropy becomes zero if the outcome of an experiment describing the location transmission into observations is unambiguous, while reaches its maximum if all outcomes of the experiment are equally likely. Obviously, it is necessary to minimize this last mentioned ambiguity. The joint entropy $H(X,Z)$ is related to the conditional entropies $H(X|Z)$ and $H(Z|X)$ through equations (3.23) and (3.24), where the former describes the equivocation and the

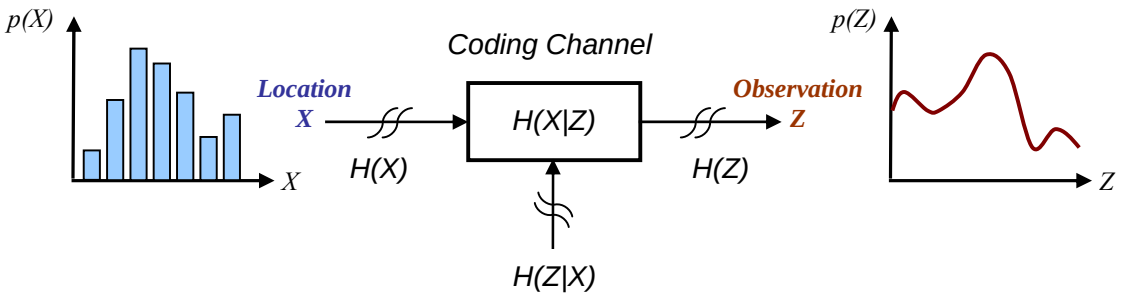


Figure 4.1. Structure of a coding communication channel. X is a random variable representing the location. Z is a random variable describing the observation domain. The location and observation variables and corresponding spaces are statistically related through the probabilistic channel, where H denotes the entropy.

latter describes the channel dissipation. The interpretation of the five entropy quantities for the general recognition process is important to understand and is summarized in table 4.1.

Table 4.1. Entropy quantities and their interpretation in the context of a coding channel definition.

Entropy Quantity	*Interpretation*
H(X)	Average information per location x_i; $1 \le i \le n$; n is the location distribution size
H(Z)	Average information per feature or observation z_j; $1 \le j \le m$; m is the feature or observation distribution size
H(X,Z)	Overall uncertainty of the recognition
H(X|Z)	Equivocation; indication for the overall quality of recognition. The smaller the value, the better that the location variable X be recognized through the measurement variable Z
H(Z|X)	Dissipation; indication for the average 'noise' or error of recognition. The smaller the value, the better that the location variable X be recognized through the measurement variable Z

The joint and conditional entropy quantities stated in table 4.1 all form possible definitions for the channel, since they provide measures to judge the quality of the recognition. We choose, however, to model the environment based on a defined transmission property. It is assumed that the environment is regarded as a set of properties which are monitored by the observations. Each property emits a certain virtual transmission value. Higher transmission values signal high distinguishness of the property, and hence relevance

for incorporating in an environment map. Lower transmission values indicate lower distinguishness and relevance and consequently non-efficiency for incorporating in a map.

Therefore, the transmission or transinformation quantity (equation 3.25) is selected to quantify the relationship between the location and observations spaces. The quantity indicates how much the observations contribute to recognition of locations and hence should be maximized. Transinformation is also defined by the decrease in uncertainty (equation 3.27), and is related to the conditional entropy by:

$$I(X,Z) = H(X) - H(X \mid Z) \tag{4.1}$$

$$= H(Z) - H(Z \mid X) \tag{4.2}$$

To maximize *I(X,Z)*, either equation is solved by maximizing the unconditioned entropy while minimizing the conditional entropy. The a priori probabilities of location, *p(X)*, are normally assumed to be known or assigned uniform distribution indicating maximum entropy (i.e. occurrence of x_i is uniformly distributed). Therefore, *H(X)* is constant, while all the other entropies change with respect to the observation *Z*. Consequently, it is easier to follow equation (4.1) for the transinformation maximization, and reduce the problem to minimizing *H(X|Z)*, assuming *H(X)* is given. Therefore, the coding channel is finally defined by this conditional entropy, as indicated in figure 4.1. The channel still maintains the relationship between the assumed input and output entropies as defined by equations (4.1) and (4.2).

A slight modification still needs to be performed. The conditional entropy, *H(X|Z)*, quantifies the feature extraction process as a whole. What is actually desired is to quantify the transinformation between every single property (a property corresponds to a sample z_j of the observation space) and the location variable. This is a quantification for the actual contribution of every feature to the localization in terms of the utility for categorization. Therefore, it is desired to calculate $H(X|z_j)$, and the objective function becomes searching for the z_j values minimizing $H(X|z_j)$:

$$\arg\min_{z_j \subset Z} H(X \mid Z = z_j) \tag{4.3}$$

Supposing that the environment space is decomposed into *n* distinct locations $x_1, x_2, \ldots, x_n$, and the observation space is downsampled to a distribution over the features $z_1, z_2, \ldots, z_m$

as shown in figure 4.1, then the entropy evaluating criterion which quantifies the relation between the space variable and the corresponding distorted observations becomes:

$$H(X \mid z_j) = -\sum_{i=1}^{n} p(x_i \mid z_j) \log_2 p(x_i \mid z_j) \tag{4.4}$$

Equation (4.4) will be set as an evaluating criterion to filter environment properties that make the location identification minimally ambiguous. The interesting aspect about using the coding channel is the possibility to evaluate different measurements of either different modalities or different feature extraction methodologies, which is considered an evaluation for the employed sensors and feature extraction methodologies.

In conclusion, we refer that building an environment model is mapping from a robot's position to sensor values. In order to fit proper and exact sensor information for localizing the robot accurately, this mapping is inverted in a trial to find the optimal mapping which maximizes the accuracy of robot location estimation. This consideration has been described via an information-theoretic formulation, in which the environment is modeled based on a domain of properties, with every property delivering certain measurable information. This information is quantified through the transmission value which has been reduced to the reduction in entropy value. The higher the reduction value, the higher the indication of usefulness of corresponding property to resolve the ambiguities, and hence relevance for employment in the environment map.

4.3 Design Aspects for the Solution

As introduced in chapter one, an important requirement for robotic systems is that they should be implemented with minimum complexity. Such issue is crucial for systems that constitute hierarchy or nestedness as the one in figure 1.3, and those involved with real-time computing constraints. Therefore, time and space complexities are necessary system considerations. This does not imply that a lower complexity is set on account of accuracy. A high or at least an acceptable accuracy level is a mutual requirement to guarantee a satisfactory performance.

For this sake, the information-theoretic modeling approach has been proposed. The argumentation about its employment has been given in section 1, and a view of its application procedure has been explained in section 2. The modeling approach generates an information-rich map which assigns the localization with desired accuracy and computational complexity

performance combination. Nevertheless, further enhancements regarding complexity can still be achieved. For example, minimal data representation and suitable organizational structure of system components can augment additional complexity reduction.

Figure 4.2 defines three claimed efficiency measures for environment model building and localization to be achieved (higher localization accuracy, lower complexity and increased robustness). The figure also shows how these measures are realized in this chapter. An environment model should satisfy a maximum bound for the localization accuracy with minimum computational complexity. The application of information theory satisfies these requirements by fitting only the exact amount of information needed by the localization task. Additional complexity reduction is afforded if data can be stored in compressed form (e.g. dense vision data allow this possibility). Therefore, a codebook module is proposed for minimizing data representation via compression, which supports further complexity reduction. The codebook module may induce a slightly decreasing effect on the task's accuracy due to partial loss of information. That is why it is not connected to the localization accuracy as shown in figure, though the mentioned degradation effect can be negligible.

One more necessary requirement for the efficiency of the solution is robustness. The solution should be robust with respect to several issues, such as feature detection, recognition and environment variations. The robustness requirement is fulfilled through machine learning and local feature extraction. Such choice maintains the robustness of feature detection and recognition under different feature transformations and distortions, as well as the robustness of location identification under different illumination conditions, clutter and partial occlusions and, more importantly, operating environment dynamics.

Some aspects have been regarded in the design of the solution, such that they fulfill the previously-mentioned requirements and comply with the used methods and concepts. The information-theoretic modeling approach can possibly be applied on the topological level or the metric level. Preference is set to its application on the topological level for various reasons. Topological modeling practically underlies many flexible capabilities for the robot, most notably with large-scale self-localization and navigation. This is generally suitable for mobile robots, and specifically for many service robots which are wide-spread nowadays. Service robots may need to have more understanding of the spatial and functional properties of the environment in which they operate. An environment model acquired by the robot

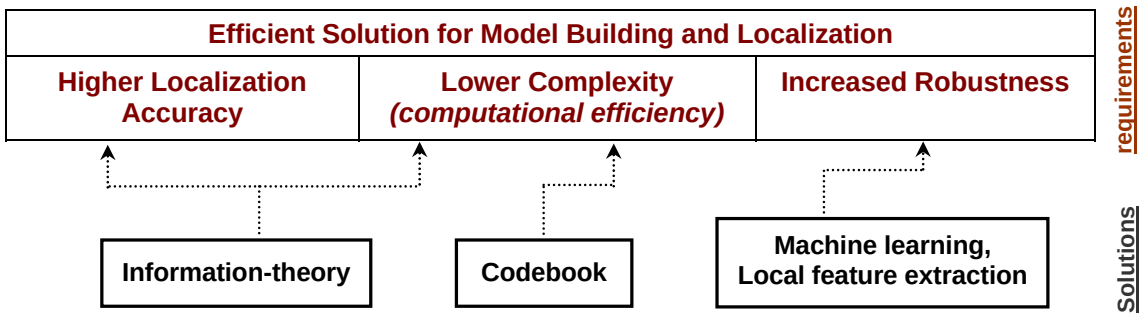


Figure 4.2. Efficient solution for environment model building and localization. An efficient model should provide robust and accurate localization with minimum computational complexity. The outlined proposed solutions contribute to those requirements for topological model building and localization.

should be consistent, and preferably human-compatible. Since a human perceives space in terms of high-level information (such as scenes and objects, and in other abstract terms as states, descriptions and relationships), it is required that mobile robots support the encoding of information the same way. Such requirements are afforded easily through the scalable topological model which maps the environment as discrete places with interconnectivities.

Furthermore, the application of the information-theoretic modeling at the topological level is more effective. This is because the processed space is smaller than the metric space with the fact that more data are available at the topological nodes. Such issue reduces the effect of aliasing and correspondence problems with better possibility for local data compression. Moreover, this design choice subsequently allows the approach to be extended into a suitable organizational structure (hierarchy), providing hybrid navigation capabilities (topological and metric). The efficiency in construction of the higher topological level will support the efficiency of the lower metric level. The general application of the information theory will influence differentiation between probable environment features (objects) and places, which will resolve the robot's location at both levels.

The rest of this chapter will tackle a solution approach, summarizing figure 4.2, from topological modeling and topological localization perspective. The topological environment description and identification will be addressed, in the view of information theory application, for a highly unstructured and dynamic environment. The topological solution forms a stand-alone unit, which will be fused into a hierarchical organization to be introduced in a later chapter, for the sake of hybrid modeling and localization.

4.4 Problem Formulation and Solution Approach

General environment models, specifically topological models, suffer from a perceptual aliasing problem when employed for navigation. That is to say, they have difficulty in distinguishing adequately between the different classes' instances (e.g. locations, objects, or features) due to data similarities. This issue correspondingly creates a second problem of data association or correspondence. Additionally, the data provided by sensors are possibly huge or redundant. The result is that systems become complex, and high computational power is needed and sometimes wasted. One possible solution is to intentionally place simple visual markers or artificial beacons to assist navigation. However, the environment may not allow this change easily, and in other times, it is required to automatically construct a model based on the environment natural features and to minimize human interference.

To overcome the above-mentioned problems and difficulties, information-rich models are suggested. These models are constructed with the minimal amount of quality-based data that resolves the ambiguity, to avoid overloading the robot system with a large overhead. Therefore, it is suggested that environment places be marked as much as possible in a unique separable way. In other words, each place should be characterized by discriminative information only, which makes it recognizable among the other places. This reduces environment and perception aliasing, and guarantees higher localization accuracy because the robot is distracted away from the non- or less-informative data. From another simultaneous perspective, minimum data will be stored in the model, an issue that reduces computational complexity and accelerates localization. In such a way, accuracy combined with efficient resource utilization, which is highly desired by complex systems, is achieved.

In accordance with the previous statement, the main problem related to the topological model building and localization with freely-selected sensor(s) and the solution approach are mathematically formulated as follows:

Problem Formulation:

Given: (1) A set of environment significant places $N = \{N_1, N_2, ..., N_n\}$ of size n;

(2) A mobile robot equipped with a sensor capable of capturing a pattern of multiple features for every place $N_i, \in N$; $i=1,...,n$, or a combined sensor configuration to fulfill this condition;

(3) Let N_i be described by a set of features S_i of potentially different size per place. The whole feature space is denoted by F so that $S_i \subset F$.

Constraints:
(1) No a priori specification about the environment (objects, landmarks);
(2) No a priori knowledge about previous positions of the robot;
(3) The environment possesses a moderate or high amount of details;
(4) Scene dynamics and varying illumination environmental influence.

Required:
(1) Construct an environment topological model $\mathbf{M}^t$ (N,C, f^*) in the form of undirected graph $T:=(N,C)$ (see figure 4.3-b), where N forms the graph nodes, and C is a set of ordered pairs indicating the spatial interconnection between nodes N_i and N_j; $i,j=1,...,n$; $i \neq j$. $\mathbf{M}^t$ comprises a minimal feature set per place $f_i^* \subset S_i$, which corresponds to a total minimal feature set in the model $f^* \subset F$; $f_1^* \cup f_2^* ... \cup f_i^* ... \cup f_n^* = f^*$, such that the general node identification probability $p(N_i \mid f_i^*)$ is maximized;
(2) Find the most probable current position(s) of the robot p^t; $p^t \in N$, using $\mathbf{M}^t$ and the extracted data from the same employed sensor(s) only.

Solution Approach:

The solution approach is based on a generic information-theoretic appearance-based modeling, which does not adhere to a specific sensor type, type of feature extraction, or precise characterization for the environment. Figure 4.3-a explains the appearance-based modeling. It shows a sketch for a part of an indoor environment where every place is represented by a few data set (e.g. images) under different acquisition or environment conditions (e.g. illumination, dynamic objects). A topological model (figure 4.3-b) is constructed, preserving the neighborhood and/or the distance information, as well as the corresponding signatures or characterizing patterns of the data set in every node. Such type of modeling gives robustness for the model, since various dynamics influencing the environment are automatically mapped in it. Other than the basic components of an arbitrary feature extraction and a matching procedure, the following three essential components are presented:

(a) *Optional* **preprocessing component** for isolating the influence of disturbances (e.g. noise and intruding features) from the data *(i.e. robust features).*

(b) **Information-theoretic evaluation component** for recognizing and filtering the most discriminative feature set which records maximum transmission to the topological places, and in other words the set maximizing the node recognition probability *(i.e. information-rich model).*

(c) **Codebook component** for compressing the filtered feature set to generate the map with the minimal possible representation *(i.e. compact model).*

Although the problem formulation has been introduced to construct a complete topological model, consisting of nodes characterized by patterns plus interconnections, only node construction is targeted in this work scope. The interconnectivity, which is normally induced through odometry measurements, is excluded from the provided solution. Therefore, the efficiency of the solutions with respect to transition information is not a part of the work.

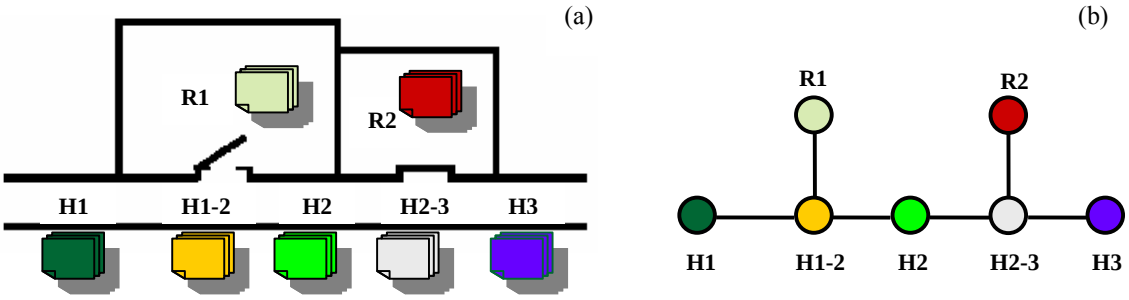


Figure 4.3. (a) Environment sketch with an appearance-based modeling. Each place is characterized by a few pattern set. (b) Generated environment topological model.

4.5 General Structure of the Solution

The information-theoretic-based solution for a topological environment structure is based on three particular processes: feature extraction, feature evaluation/selection, and feature compression. The topological node – *which is equivalent to an environment place* – will be characterized primarily through robust feature extraction. To induce an information-rich model and improve the computational performance of topological localization, supervised features evaluation and selection are applied in the following step. A final process concerns the proper storage formatting for the evaluated features. It applies feature compression to generate a compact model and accelerate the localization performance once more.

The first two processes of feature extraction and evaluation/selection follow the procedural steps of designing statistical pattern recognition systems, as proposed in [Duda et al., 2001]. Figure 4.4 shows a similar suggested procedure to build the environment representation based on the evaluation of extracted features. Blocks in bold are main modules that are common in any pattern recognition system, while shaded blocks are the modules proposed for the solution design. The procedure depends on feature extraction and machine learning, and the goal is to characterize a given object (i.e. an environment location in our case) by discriminating measurements or features, whose values are very similar for objects in the same category and at the same time very different for objects in different categories.

The procedure flows in two phases: an offline phase for training and an online phase for recognition. Four blocks out of five are common in both phases: sensing, pre-processing,

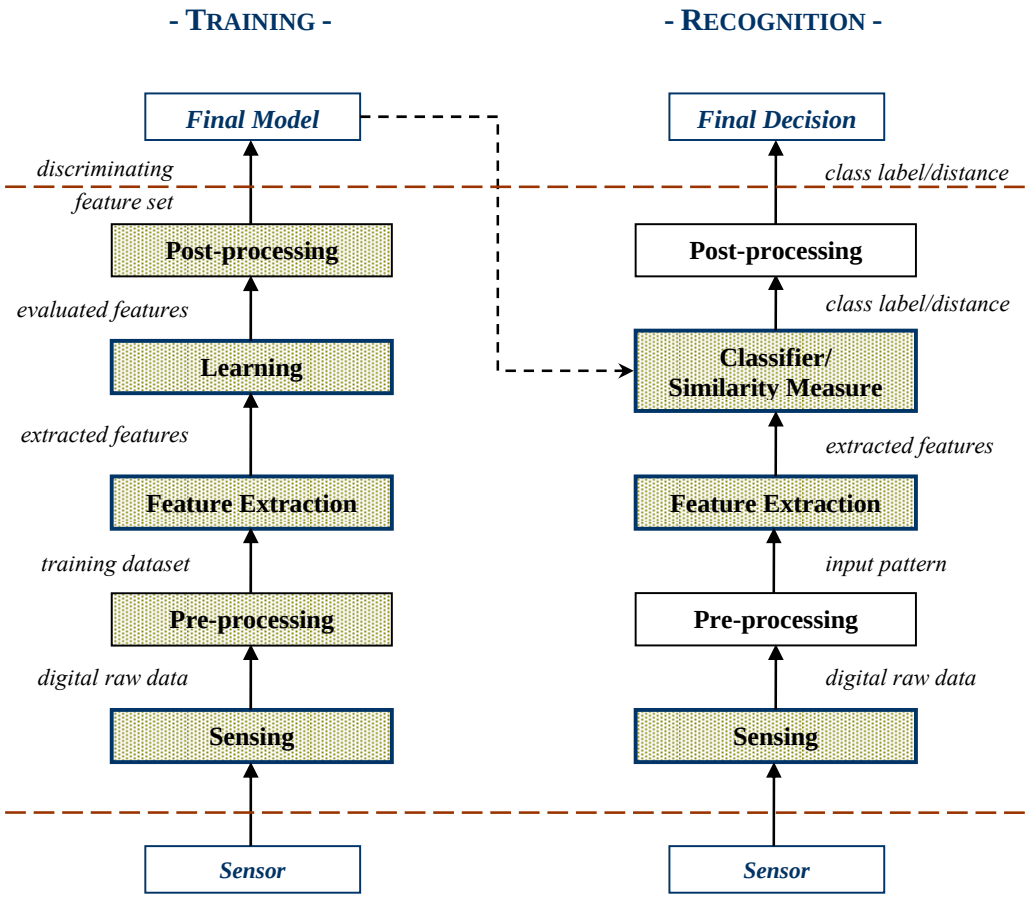


Figure 4.4. Suggested learning procedure for feature evaluation. The procedure follows the general design of pattern recognition systems, and aims at extracting a set of robust and discriminating environment features.

feature extraction and post-processing. The sensing block perceives the environment space by capturing data in raw format. An optional preprocessing simplifies or filters the captured data for subsequent operations, while trying not to lose the significant information. Data are reduced by the feature extraction process through measuring and extracting certain 'features' or 'properties'. It is important that those features be insensitive to several variations (e.g. transformations). The process provides a new representation for the input pattern resulting in simplification of the model. Finally, the post-processing block uses the output of the learning process to decide on recommended actions for the generation of a final model.

Learning is the core of the training phase. It evaluates the evidence at hand and generates a pre-final statistical model. The output of learning is a set of evaluated features as outlined in the figure, which is acted upon to produce the final model. A learning block can be a method for training a classifier or for feature selection. It can additionally be supervised, unsupervised, or reinforced. The recognition phase proceeds the same way as the training phase, except that the learner is now replaced by the final model. The model is often termed a classifier, in which a class label for the input pattern is generated, and sometimes accompanied with a related distance measure.

As reported in the state-of-the-art, few techniques focusing on the quality of features do exist in the literature of robotic map building and localization. The massive features' pruning and dimensionality reduction techniques have been tackled seldom, among which some were unsuccessful [Ayers and Boutell, 2007]. We believe that compression based on selected features using a quality measure is the missing concept in the previous unsuccessful work. For this purpose, and from the human notion of recognizing places and defining distinguishing features, the introduced information-theoretic evaluation criterion (4.4) is placed at the learning block in supervised procedure. The idea behind employing the criterion is the cognitive human-driven concept that items or objects which are equally distributed among all class categories are non-informative objects. On the other hand, less frequently encountered items reveal more information about identity and membership. An ideal case, though not in practice, is that each class maintains uniquely distinguished object(s). Therefore, the suggested information-theoretic-based learner will make use of this concept by studying the properties of the environment statistically and evaluating them.

The suggested supervised modeling of figure 4.4 is projected into a conceptual design and realization for the robotic model building and localization solutions, as shown in figure 4.5. The design follows the sequential processes – feature extraction, evaluation and compression – before generating a final feature map form as shown in figure 4.5-a. Detailed structure of the conceptual design is introduced in figure 4.5-b. Figure 4.5-b is explained in two separate phases: model building and localization. Model building is executed in the initial environment exploration phase. It includes the following modules: (1) *Feature extraction* module related to the employed sensor or combined sensor configuration; (2) *Preprocessing* module for detecting and eliminating outliers; (3) *Information-theoretic evaluation* module for weighing the features according to their node discrimination capability measured by their transmission values. The module contains a selection function to filter the features into a reduced set according to a certain threshold. Additionally, it contains an implicit processing functionality that prepares the data for proper compression. The output of

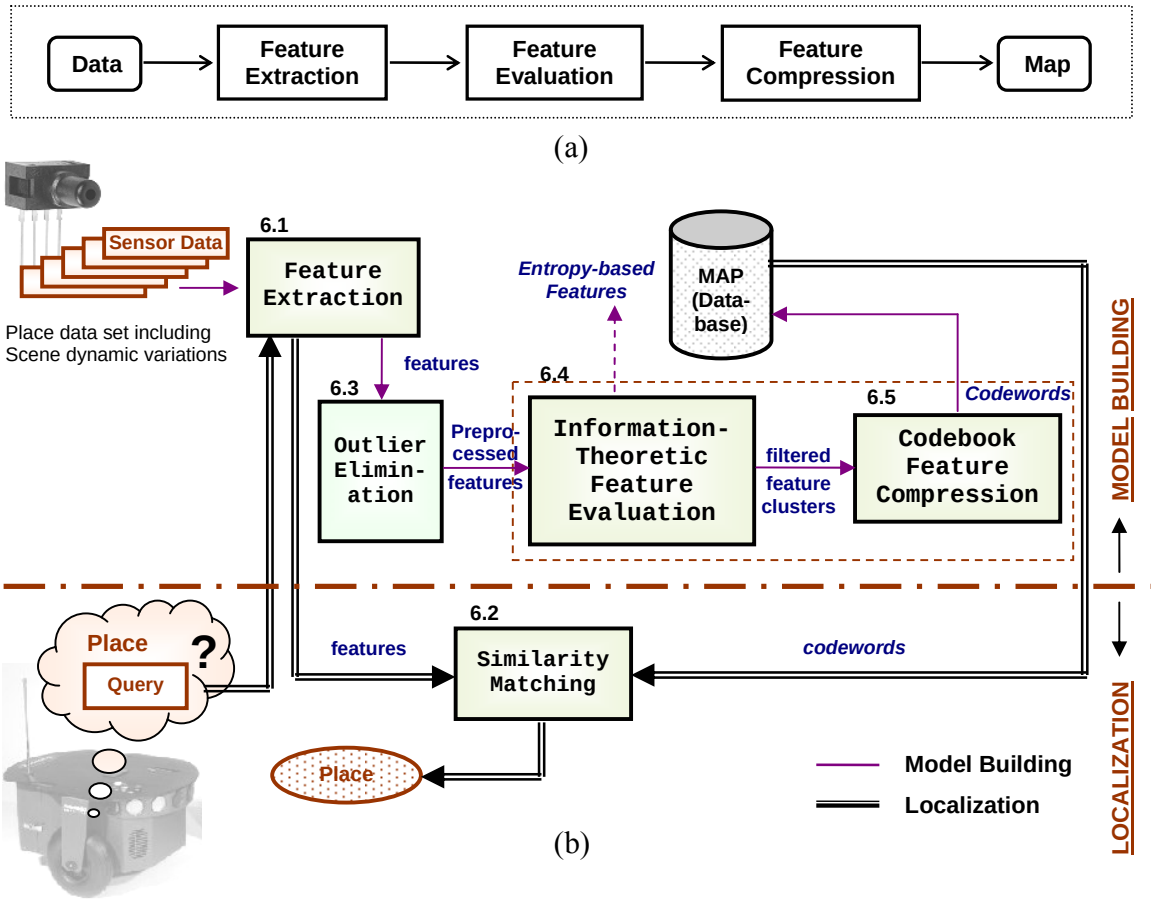


Figure 4.5. (a) Model building and map generation concept. (b) Proposed realization and solution structure.

this module is two versions of information: the evaluated features which are termed *entropy-based features* and a corresponding feature cluster identifier for every evaluated feature; (4) *Codebook* module for compressing the feature clusters to produce *codewords*. The codebook accounts for a final feature-map representation with the codewords accounting for the map features. Both the information-theoretic evaluation and codebook can be regarded as a unified block that generates two data formats: low-resolution data (i.e. the compressed codewords) and high-resolution data (i.e. entropy-based features). Robot localization is a process of querying the map. The current pattern sensed by the robot is compared to the map using a distance measure (the similarity matching block), and the result indicates the most probable place corresponding to a topological position in the map.

The solution structure in figure 4.5 is a general structure that fits metric localization as well as topological place recognition. This depends on the type of feature characterization that induces either a topological or a metric feature map. As previously mentioned, the final design targets the construction of a hybrid map that will contain both types of features. In accordance with the design of this chapter, the topological feature map is discussed, while a modified structure for metric solution adaptation will be introduced in chapter six.

The proposed solution structure includes the following three advantages. First, it is generic because it is information-theoretic-based, and hence applicable to different sensor modalities, feature abstractions and measurements. Places are not restricted to appearance characterization through vision or range measurement features (e.g. corners, openings) only. They can also be characterized by surface reflectance, tactile properties, temperature, sounds …etc, or a set of combined features which is highly recommended. The second advantage is that environment modeling using natural features or landmarks is efficiently represented in such solution structure. Features or landmarks can be extracted and filtered automatically without having to be explicitly identified. The third advantage is that it is robustly designed, since an employed appearance-based modeling implicitly embeds several factors that may influence the localization performance.

4.6 Structural Components

The different structural components of the solution, which are outlined in figure 4.5-b, will be discussed in details in the following subsections.

4.6.1 Feature Extraction

Adopting a specific feature extraction method usually depends on the available sensor. In the application point of view, any sensor(s) along with a suitable feature extraction method(s) matches the proposed solution. Nevertheless, for a uniquely employed sensor, a single condition should be satisfied. It should be capable of acquiring multi feature measurements in the same topological place. This condition is also coupled with the robot's navigational behavior. An example of an unsuitable sensor-behavior is a range sensor mounted on a robot, whose behavior is restricted to free space or wall following in a corridor-like environment. In such behavior, the sensor provides only constant relative distance which does not build a signature for the topological node. Examples for sensors that fulfill the mentioned requirement are vision sensors and scanning laser rangefinders. They can gather a bundle of measurements, almost of non-uniform distribution. A set of combined sensors relaxes the single sensor constraint, since the place will possess several features that indicate a total measurement variation and a pattern probabilistic distribution.

It is worth mentioning that the solution approach is not successfully applicable if the environment does not possess enough details. That is because the approach measures the transmission of every environment property in order to eventually pick up the smallest informative set. Therefore, an already existing pool of properties or features should be available. This is what is meant by environment details. In the case that combined sensors and/or feature extractions algorithms are used, it is the solution-approach's role to quantify both sensors and features through the assigned information-theoretic measure to evaluate their contribution to minimizing the uncertainty.

Although the choice of the sensor, feature extraction and modeling methodologies are arbitrary, some preferences are recommended. Practically, it is encouraged to employ *vision* as a main sensor. Vision possesses attractive advantages by providing rich capture and characterization abilities for the environment with passive interaction. They are additionally now affordable with low-costs and low-power consumptions. In conjunction, *an appearance-based modeling approach* is recommended for the outlined advantages provided in section 3.3. Appearance-based modeling excels over other shape and region-based modeling approaches which are strictly environment specific or require exhaustive preprocessing steps. Another relevant advantage is that it is easy adaptable to learning systems.

A third and final recommendation is related to the feature extraction method. *Local interest point extraction* is recommended for the non global application on the sensed pattern. Global feature extraction application may be a burden or even misleading when the important information is confined in a small local region. The literature provides several examples of local features, such as the SIFT and SURF explained in 3.4.2. Both methods show outstanding performances, especially in object recognition. The SIFT feature extraction is chosen since it is a dominating choice reported in several recognition work. The descriptor is based on characterizing interest point by non-geometric gradient information (see section 3.4.2.2). Non-geometric characterization is a reasonable choice, because when it is additionally combined with geometric characterization, highly decisive information can be generated.

Concluding the suggested implementation, SIFT feature extraction is proposed combined with an appearance vision-based modeling. Appearance-based modeling has been previously introduced in figure 4.3-a, where every place is represented by a small set of image signatures. Few signature set is involved, instead of a single one, because the same scene is subject to (slight) variation under varying conditions leading to (non-major) variance in their generated descriptors. The few set, which is needed by the information-theoretic learning anyhow, increases the recognition robustness against such variation. Figure 4.6 shows the local interest points extracted by SIFT algorithm for a panoramic image example. The image has 913 interest points generated by the algorithm which are described by gradient descriptors. The number of extracted features is relatively small compared to what is stated by Lowe [Lowe, 2003]. This is because place feature extraction from a far point can induce fewer features in comparison to a closer point. Nevertheless, the number of local features is still high and will be reduced simultaneously through the proposed feature evaluation, selection and compression modules.

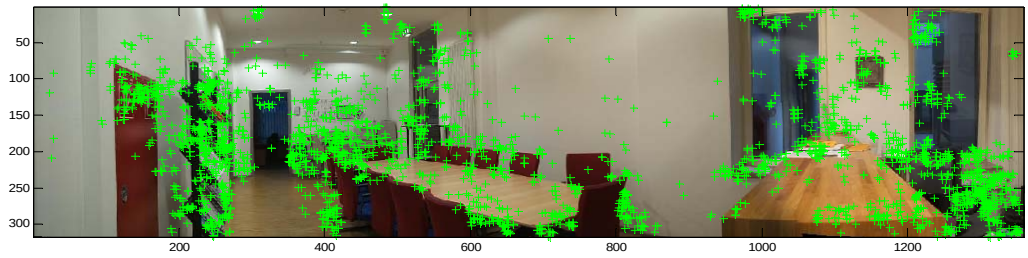


Figure 4.6. An Indoor panoramic image with features extracted by SIFT algorithm (No. of features= 913).

4.6.2 Similarity Matching

The similarity matching module compares the features extracted from the image viewed by the robot with the features stored in the map in order to infer the robot's topological location (i.e. topological node). Some issues are noted in the context of similarity matching. Firstly, not every feature is always detected by the sensor. It may be missed due to changes in the robot pose, view point, illumination conditions, shadows, occlusions or sensor noise. Secondly, some undesired features may arise because of the natural clutter and dynamics that occur in real-world environments, such as scene backgrounds, newly added objects, moving people and vehicles, or other external imposed factors (e.g. illumination conditions, flashing lights effect). That is why learning is a good strategy for increasing the robustness against those factors. However, such factors cannot be completely avoided. Therefore, what is required is a choice of robust matching criteria that can eliminate the false feature matches and balance the matching for the partially detected features.

The best candidate node match for a given sensed pattern is computed by comparing the pattern with the map using features majority voting. This requires measuring the distance between each feature in the sensed pattern and those of each node in the map then counting the corresponding feature matches found. On completion, the percentage of the total number of matches relative to the number of features of the sensed pattern decides for the best candidate node. For selecting the most accurate corresponding feature match, selective matching is applied by imposing a criterion. The criterion compares the distance of the first closest neighbor d_1 to that of the second closest neighbor d_2 for every single feature matching. If the ratio is greater than a given threshold *thr* [Lowe, 2004],

$$\frac{d_1}{d_2} > thr \tag{4.5}$$

then the feature is not accurate enough and is discarded from voting. For closest neighbor search, the Cosine angle distance, d, is suggested:

$$d(A_i, B_j) = \cos\theta = \frac{A_i^T . B_j}{\| A_i \| \| B_j \|} \tag{4.6}$$

where A_i is the l-dimensional vector of the i^{th} feature extracted from the current view and B_j is the l-dimensional vector of the j^{th} feature in a given map node, $||A_i||$ and $||B_j||$ are the

corresponding norms of the vectors defined by:

$$\| A_i \| = \sqrt{(\sum_{k=1}^{l} a_k^{\,2})}; \qquad \| B_j \| = \sqrt{(\sum_{k=1}^{l} b_k^{\,2})}; \tag{4.7}$$

The Cosine distance gives a score in the range of [0,1], where zero indicates completely dissimilar vectors, while one indicates identical vectors. It acts the same as the popular Euclidean distance in high-dimensional data spaces of Content-based Retrieval systems [Qian et al., 2004], and has the advantage of being faster in computation.

4.6.3 Outliers Detection and Elimination

Outliers exist in every data gathering process. They always have a fraud effect that negatively affects systems' accuracy. Hereby, a preprocessing filtering step is beneficial, especially in learning systems. Outliers in this sense represent unstable features in patterns of the same place. Stable features tend to appear in every extracted pattern, independent of illumination or dynamics of the environment. On a higher representative level, though not ideally extracted with current automatic outlier detection techniques, outliers can be features acquired through lighting conditions (e.g. shadows, a camera flash), or objects that temporarily intercept then vanish from the scene. For example, a camera flash or light reflection on a surface could be in fact distinguishing, but undesired, feature. An outlier can also be produced from a dynamic object that temporarily interrupted the scene when sensed by the robot during data gathering. The goal then is to eliminate those features that reduce map quality, and may lead to potential misrecognition. Eliminating outliers has a better effect regarding class consistency, primarily, and features size reduction, secondarily. Outlier detection is often based on distance measures, clustering and spatial methods. Several approaches have been investigated for outlier detection [Chandola, 2007], among them are clustering-based methods.

Clustering is an unsupervised machine learning technique to group similar data instances together using a given measure of similarity. Clustering algorithms attempt to organize unlabeled feature vectors into clusters or groups, such that samples within a cluster are more similar to each other than to samples belonging to different clusters. Despite the advantages of the cluster-based techniques that they do not have to be supervised, in addition to being used in an incremental mode (i.e., after learning the clusters, new points can be fed in to the system and tested for outliers), the disadvantage lies in the fact that they are

computationally expensive as they involve computation of pairwise distances. Since there is no given information about the underlying structure of data or the number of clusters in clustering approaches, there is no single solution or even a single similarity measure to differentiate all clusters. For this reason, there is no theory that describes clustering uniquely. Nevertheless, the performance of clustering-based techniques is surprisingly outstanding. In [Yoon et al., 2007], the *k*-means clustering algorithm is applied successfully for outlier detection of software management data. Therefore, their long processing times can be tolerated as long as they are performed in offline learning phases.

Outlier removal via clustering relies on the key assumption that normal data points belong to large and dense clusters, while outliers either do not belong to any cluster or form very small clusters. The assumption holds true, since data in general contain lots of redundancies that correspond to a certain object or multiple occurrences of similar objects. At least, robust features will appear for the same scene in the extracted pattern set, independent of any external factor like lightening conditions or environment dynamics.

For implementation, the *k*-means clustering (see appendix A) with a Cosine similarity distance is suggested for this module. The unsupervised approach is quite suitable, because it is hard to obtain prior knowledge of a classification label for the local point features, being fraud or real, weak or robust. To avoid discarding too many features from the images of locations with few detected features, the value of *k* is set to be a function of both the number of images per place and the average number of extracted features per image.

4.6.4 Information-Theoretic Feature Evaluation

The information-theoretic evaluation component is one main improvement in environment model building, which contributes to generating information-rich maps, increasing accuracy of localization and reducing its online computational complexity. In simpler words, the function of component is to evaluate perceived data by sensors and filter data with the highest information content only. Data of the highest information content contribute to less location uncertainty and hence are selected for optimizing localization.

In specific terms, the component maximizes the *transinformation* between topological places and corresponding extracted features. This idea has been introduced in section 4.2. In this idea, each extracted feature has an associated transinformation value contributing to the

topological node identification, or rather, uncertainty reduction. The higher the trans-information value, the better the discrimination power of feature to identify the node[1]. The feature space is accordingly classified into either highly discriminating space (features possessing higher transinformation) or less discriminating space (features possessing lower transinformation). Such classification manages to find out which features give the minimum localization uncertainty and how many approximately are sufficient. A final filtering operation for the desired information (highly transinformative features and discarding the lower transinformative features) generates the information-rich map for accurate localization.

In section 4.2, transinformation quantity has been reduced to the reduction in uncertainty, measured by the conditional entropy of location given features. Assuming that the topological node is a random variable *X,* and the extracted features from the environment is another random variable *F,* features are filtered based on minimizing:

$$f^* = \underset{j\in\{1,2,\ldots,m\}}{\arg\min}\, H(X|F = f_j) \tag{4.8}$$

with f_j representing feature categorization instances (i.e. discretized feature domain) and m their distribution size. These instances are called in the component design *feature categories*.

For the general feature evaluation problem, machine learning is applied. A training dataset consisting of *{Feature–Class}* tuples is needed for the statistical analysis. From this dataset, the proposed conditional entropy of a 'Class' (i.e. *X,* the topological node random variable) given a *feature category* can be calculated. In the case *X* is used for the 'Class', the problem falls into a classical node classification problem. However, another definition for the 'Class' is adopted, which encodes higher-level information in the nodes and assesses suitability for feature compression at the same time. This is through introducing the idea of codewords. Codewords are a compressed-feature format. Therefore, in the implementation of the evaluation component, clustering is incorporated twice for quantization of features in the high-dimensional feature space, providing a two-fold: (1) a means for defining the *feature categories* f_j, and (2) a facility for defining a new intermediate data form called *feature clusters*. *Feature clusters* are defined locally for every node and are codewords candidates. The reader can refer to figure 4.5-b to see the input and outputs of the evaluation component.

[1] Transinformation is considered a measure for the distinguishability of feature, as well as for its discriminating power, to decide among the different places.

The evaluation component proceeds by firstly generating the *feature clusters*, then secondly the *feature categories*, and finally applying the filtering criterion.

To generate the *feature clusters*, features of patterns of each node are split by means of *k*-means clustering, defining new compressed versions for the extracted features. The step is augmented by the extraneous redundant features that are mostly present in an image, and which may refer to an exact, a part of or a characteristic appearance of an object. The value of the number of clusters, *k,* is set differently in every node, because there exist nodes with few detected features and others with more. Therefore, *k* is set to be a function of both the number of images per node and the average number of extracted features per image. Local clustered features are called 'feature clusters' or 'keypoint clusters' in reference to SIFT. These generated clusters form the 'Class' column or the supervisory label in the training dataset, while the 'Feature' column simply refers to the value of feature or keypoint to be evaluated. The dataset is obtained by letting the robot move in the operating environment and build the topological graph, whose nodes contain the gathered extracted features.

To generate the *feature categories*, the whole feature space is quantized to give the discrete sample space which is a measure for the actual probabilistic variation of data. The quantization is performed by applying clustering again. This time, however, clustering is applied on the whole features in the training dataset, in contrast to the prior local clustering applied on each node to generate the *feature clusters*. The output clusters are termed *feature categories*, whose distribution represents the true realization of features' variation. The *k*-means algorithm is also used here as in the prior clustering, but surely with different parameter value for *k*. The previous two steps of generating the *feature clusters* and *feature categories* are illustrated in figure 4.7.

Eventually, the conditional entropy criterion of recognizing a *feature cluster O*, given a sampled *feature category* f_j is calculated through the posterior probabilities:

$$H(O \mid f_j) = -\sum_k P(o_k \mid f_j) \log_2 P(o_k \mid f_j) \tag{4.9}$$

for every k=1,…,Ω, where Ω is the number of instantiations of the feature clusters, and j=1,…,Ψ, where Ψ *is* the number of instantiations of the feature categories. Equation (4.8) is consequently updated to searching for the minimal feature set f^* that minimizes (4.9):

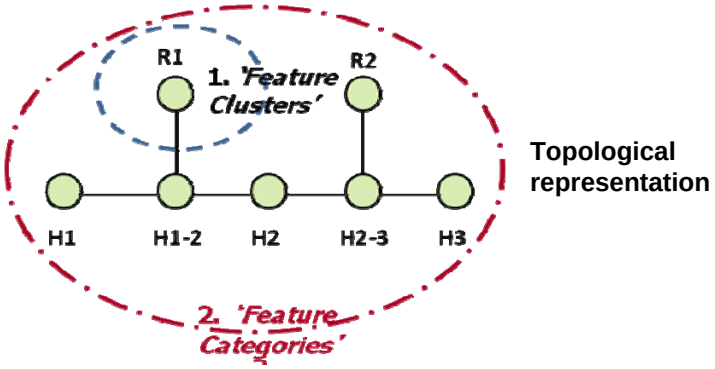


Figure 4.7. Generating feature clusters and feature categories. Clustering is applied first on the node level data to generate the feature clusters, while next on the whole map data to generate the feature categories.

$$f^* = \arg\min_{j\in\{1,2,\ldots,m\}} H(O|F = f_j) \tag{4.10}$$

The posterior probabilities $P(o_k \mid f_j)$ in (4.9) are calculated from the likelihood of *feature clusters* and the prior probability of *feature categories* through the Bayes formulae:

$$P(o_k \mid f_j) = \frac{P(f_j \mid o_k).P(f_j)}{P(o_k)} \tag{4.11}$$

Equation (4.9) distinguishes the quality of extracted features in their sampled space, as well as the created feature clusters. Features that tend to appear equally likely among the different nodes as well as feature clusters are less informative, and thus will encounter high entropy values. Those features are called *high-entropy features*. On the opposite side, features whose occurrence is bound to few nodes and feature clusters deliver more information (distinguishable in terms of categorization), and will encounter low entropy values. Those features are called *low-entropy features*. Consequently, a decision regarding each extracted feature, whether it is useful for node identification or not, can be made based on the information it delivers (i.e. if entropy is low). In a similar manner, the quality of the created feature clusters is determined through the evaluated features, and hence non-relevant feature clusters will diminish after filtering out the high-entropy features. Filtered feature clusters, which correspond to the low-entropy feature set, will be the selected codewords candidates that are passed to the compression codebook module.

An important parameter to check in this component is the parameter *k*. With the high dimension of observation domain and the lack of knowledge about existence of evidence

related to data semantics (e.g. assembling parts of an object, marginal categorization), the right number of clusters is difficult to know. Therefore, several trials are normally conducted in order to tune the best value for the parameter. Trial ranges can be set in relation to the size of data. The quality of clustering can also be verified through a quality-based criterion, such as the average Silhouette coefficient. The Silhouette value for each point in a cluster is a measure of how similar that point is to points in its own cluster compared to points in other clusters, and is defined by:

$$S(i) = \frac{b_i - a_i}{max(a_i, b_i)} \tag{4.12}$$

where a_i denotes the average distance from the i^{th} point to other points in its own cluster, and b_i denotes the minimum of the average distances from the same point to the other clusters. The value of Silhouette coefficient varies between -1 and 1. A value near -1 indicates that the point is clustered badly. A value near 1 indicates that it is well clustered. To evaluate the quality of an applied clustering, the average Silhouette values of all points, $\overline{S}$, is computed.

4.6.5 Codebook for Feature Compression

A feature codebook is a compression technique inspired by the Bag-of-words quantization methods adopted in text classification and retrieval systems. Recently, it has been introduced to the vision domain, and has shown efficiency and simplicity in its creation. Visual codebook construction is based on defining a certain visual alphabet (visual codewords). A simple meaning for the visual codeword can be a tire, a gear and a hand drive which can define a motorbike object. In the evaluation component, feature clusters have been defined locally in every node using *k*-means clustering. This allows efficient encoding for the candidate codewords, because slightly varying feature clusters may be of interest if they belong to the filtered set. For instance, in the motorbike object, a circular thick tire can be regarded as a codeword, while a relatively thinner tire can be regarded as another codeword, although both belong to the same general ‘tire’ class.

The codebook component (CB) allows the filtered entropy-based features, which are the output of the information-theoretic evaluation component, to be stored in compressed format. It is easily generated from the information-theoretic evaluation component since the latter is intelligently processed to generate clusters of the evaluated features (i.e. feature

clusters). The CB is constructed from the filtered feature clusters. A CB entry is equivalent to a single codeword and is represented by the centroid of the corresponding filtered feature cluster. This value can be simply the mean of the given cluster members. The CB relates every codeword to the reference node(s) that encompass it. It is worth mentioning that while the robot senses the environment to recognize its location, no further clustering is applied to the sensed pattern. This is to save the time consumed by the clustering in order to detect the codewords. A direct feature-to-codeword matching using the codebook is applied instead.

In figure 4.5-b, both the information-theoretic evaluation and the codebook components are highlighted as a single integrated processing unit. This unit has two output data formats: the entropy-based features and the compressed codewords. Either data can account for a feature-based map. The figure identifies the CB – the second contributing improvement for environment model building – as the feature map. Nevertheless, both data formats will be studied in the localization performance evaluation to contrast them against each other.

4.7 Performance Measure of Localization Accuracy

The average localization accuracy is identified by the system recognition rate. System recognition rate is obtained by letting the robot execute several navigational experiments, in which the robot location is queried multiples of times, and then the average of the results over the total runs is calculated. To set a measure for localization accuracy, we adopt the *Precision* and *Recall* metrics used in retrieval systems. Precision is the measure of the ability of a search to retrieve an assigned number of the most relevant items (usually images or documents), and is defined by:

$$Precision = P = \frac{no.\ of\ relevant\ items\ retrieved\ by\ a\ search}{Total\ number\ of\ items\ retrieved} \tag{4.13}$$

Recall is the measure of the ability of search to locate *all* relevant items in the corpus which should have been retrieved by the search, and is defined by:

$$Recall = R = \frac{no.\ of\ relevant\ items\ retrieved\ by\ a\ search}{Total\ number\ of\ existing\ relevent\ items\ in\ corpus} \tag{4.14}$$

Precision and Recall measures are complementary. The difficulty resides in having both at maximum. An ideal case is when all the stored patterns representing a place are retrieved

as top matches. This is however far from practice. From the definitions of both performance measures, it can be claimed that Recall is more interesting for document retrieval, while Precision is more relevant for image retrieval applications. Nevertheless, from a point of view that Recall should be at maximum as important as the Precision, a score function is defined to be optimized to select the best combined Precision-Recall ratio.

The possible Precision-Recall combined ratio is investigated using a single score; the *E-measure*, which is defined as [Lewis, 1995]:

$$E = 1 - \frac{1}{\alpha(1/P) + (1-\alpha)(1/R)} \tag{4.15}$$

where R is Recall, P is Precision, and α is a ratio parameter. (4.15) can be rewritten as:

$$E = 1 - \frac{1+\beta^2}{(\beta^2/R) + (1/P)} \tag{4.16}$$

with $\beta \in [0,\infty]$ being the required parameter to be found, and which measures the relative importance of Precision to Recall. It is related to α by $\alpha = 1/(\beta^2+1)$. A bigger value indicates more emphasis on Recall, while a smaller value indicates more emphasis on Precision. The lower the value of E, the better the performance of Precision versus Recall. Therefore β is found by minimizing E.

4.8 HEID Test Environment, Data Acquisition & SIFT-Map Construction

The test environment is the indoor office environment of the Automation Department at the University of Heidelberg, whose dataset will be abbreviated as HEID (HEIdelberg Dataset). Figure 4.8 shows the floor plan of the department which spans an approximate area of 600 square meters. Some areas are selected for the experimentation of model building and localization. They are identified by colored boundaries as shown in the figure. The selected space constitutes 7 places that account for the map topological nodes. The space spans 5 office sections in 3 rooms (2 double office rooms and one single office room), a meeting hall with a joint kitchen and an exit/stairs view. The total selected metric space is approximately 120 square meters. Table 4.2 describes the selected space identified in the previous figure with the area of each room or enclosed space.

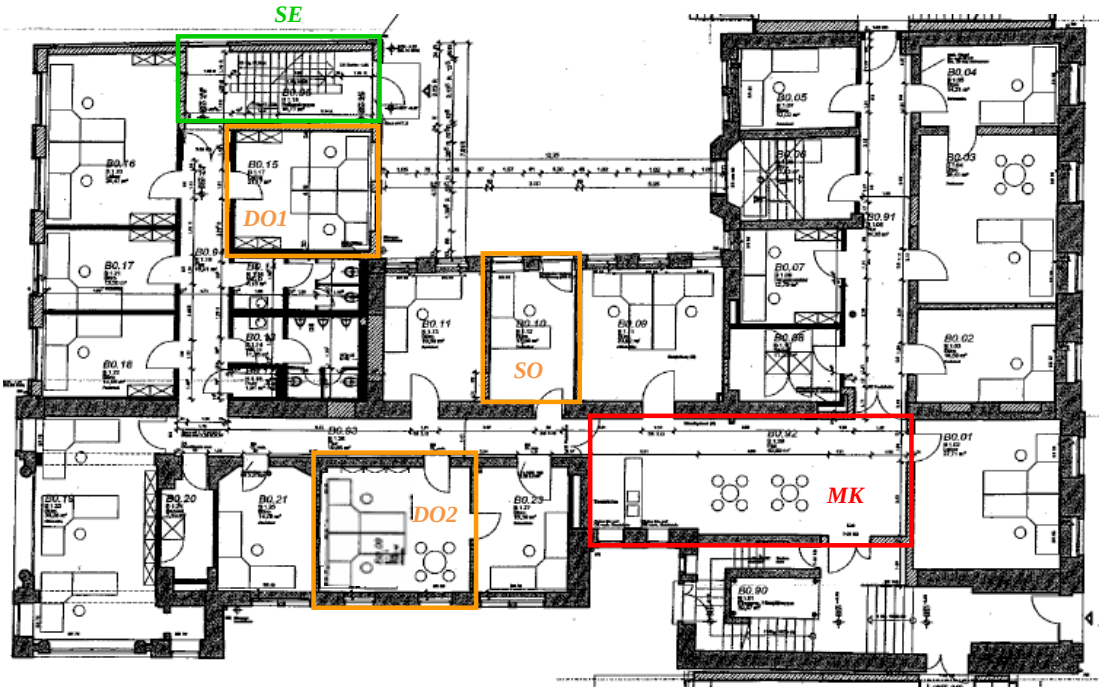


Figure 4.8. Floor plan of the Automation department at University of Heidelberg. The colored areas represent the space used for environment model building. Orange marking encloses office rooms spanning 5 office desks, red marking encloses a meeting hall and a kitchen, and green encloses an exit-stairs view.

Table 4.2. Selected space with corresponding areas. Test environment-University of Heidelberg (HEID).

id	*Room*	*Area (m^2)*
DO1	*Double office room 1*	21.5
DO2	*Double office room 2*	24
SO	*Single office room*	12.5
MK	*Meeting and Kitchen area*	46
SE	*Stairs and Exit view*	16

The environment is explored for data acquisition and an initial map is constructed containing features of the places as extracted by the original feature extraction algorithm (SIFT). This SIFT-map will be the reference for comparison while evaluating the proposed maps. It will act as the training dataset for the information-theoretic evaluation and will also be used to identify the parameter of the localization accuracy measure. The creation of this image dataset has been made using a digital Canon camera of resolution 640x480. The camera has been mounted on a tripod and images are acquired from the seven places. For each place, 5-7 panoramic images are acquired at different time intervals varying during daytime to have different illumination conditions, as well as on apart timing intervals, where the order of the

place is changed through moved, missing or newly added objects. That is to say, illumination and scene dynamics are the factors affecting the environment. The whole dataset is collected over a period of 6 months to ensure meeting those variations.

Panoramic views are constructed for places to capture a large amount of the environment details, and hence provide richer characterization. The position of the camera is slightly changed each time the same scene is acquired. This is not a severe problem since positional data are not preserved with the acquired image in our case. The wide-view images are constructed by stitching several sequential image shots for each place together (see appendix B). In the data gathering phase and initial map construction, a panoramic tool is used to perform the stitching. In the online localization, it has been automated. The tool depends also on SIFT points recognition for stitching, finding correspondences and finally creating the panorama. Each generated image describes a field of view for the scene between 180-270 degrees. The restricted field of view is due to the limitation of software capabilities. The stitching mechanism is adopted emulating wide angle cameras, such as panoramic (360 degrees) cameras and Eyefish cameras, which can perceive large parts of the environment. The emulated large field of view captures more details about the environment, and enables a better recognition with less false negatives (mismatches). Figure 4.9 shows panoramic image examples for the different topological places in the operating environment, with the classification on the right-hand side identified in table 4.2.

SIFT feature extraction is applied for the panoramic views. The extracted patterns are associated with the corresponding nodes and stored in a structured file as the initial SIFT-map. The map includes topological features of 41 wide-view images which have been acquired in six environment exploration phases. The number of SIFT keypoints detected in an image varies between 500 and 1700, with an average of 1000 keypoints per image, each with 128 dimensions. This yields a topological map of 40 Megabytes in size. Figure 4.6 shows an image example for the meeting place, with 913 keypoints extracted by the SIFT algorithm.

4.9 Experimentation and Results

The experimenting robot is a Pioneer P3-DX base platform provided by MobileRobots Inc. (ActivMedia Robotics). The platform is a two motor differential drive robot with two 19-centimeter wheels and one rear caster wheel. It is equipped with a ring of 8 forward ultrasonic

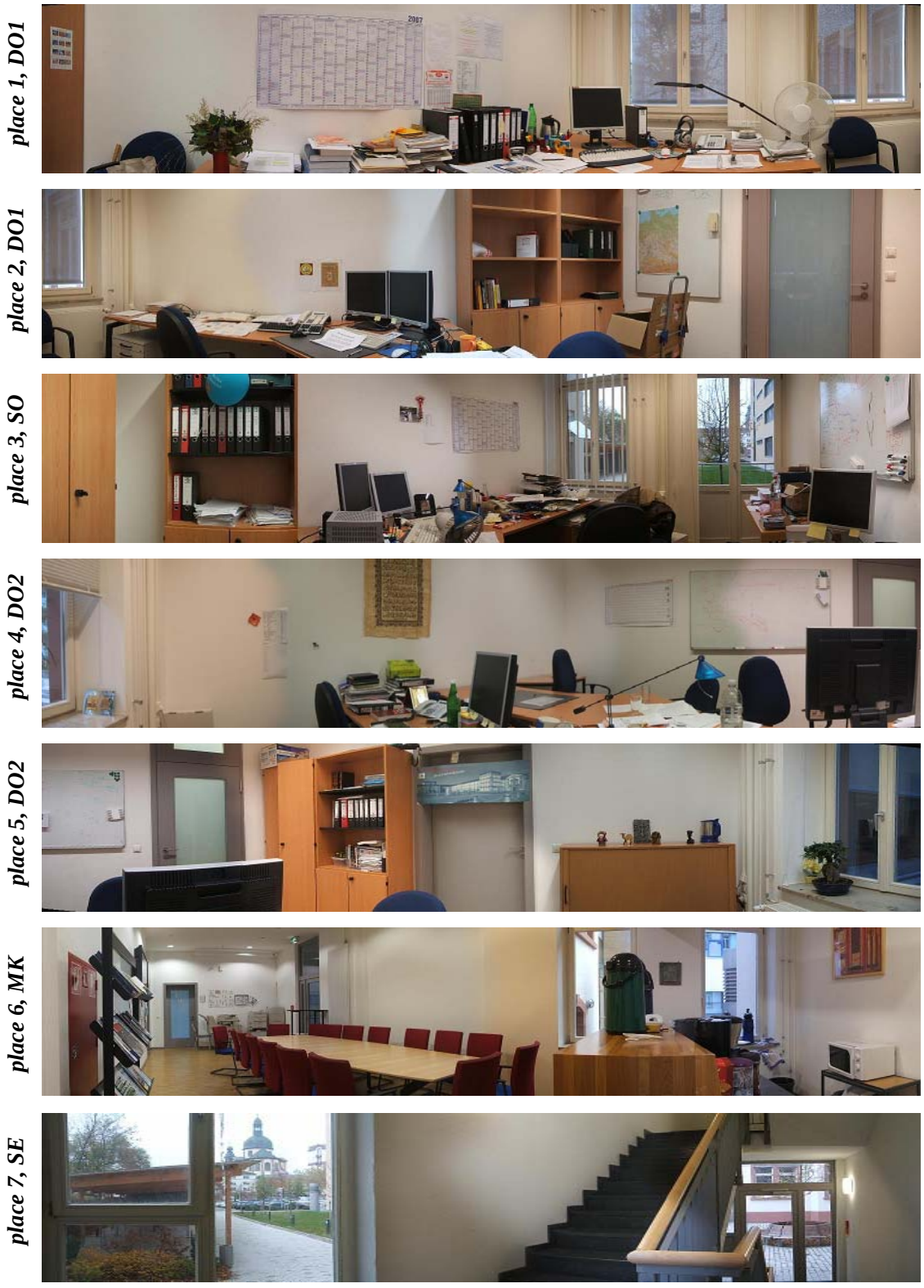


Figure 4.9. Panoramic image view examples for the selected places.

sensors, which is engaged with a single behavior for collision avoidance. A structure has been built up and fixed on the robot in order to support a camera and a laptop with windows and Matlab developing environment (Intel core Duo 2.4 GHz, 3Gbytes RAM). The mounted camera is a Logitech 4000 pro webcam of 640x480 resolution. Figure 4.10 shows the robot platform used in the experiments for evaluating the localization using the proposed entropy-based features and codewords maps. In what follows, the performance using the original SIFT algorithm is first registered, followed by the application of the proposed solution approach, and the recording of the obtained performance results.

Figure 4.10. Pioneer P3-DX mobile robot with built-in structure supporting a laptop and a webcam.

4.9.1 Feature Extraction Performance

This section records localization performance of the initial SIFT-map (i.e. localization accuracy using normal SIFT algorithm). Additionally, the algorithm's robustness is tested against noise, camera resolution and scaling. In [Lowe, 2004], the repeatability of keypoints detection in image is tested as a function of pixel noise. Here, the robustness of recognition is tested with respect to noise, lower quality sensor (i.e. resolution) and scale. This can help to identify the different conditions in which the feature extraction can still be adequate.

Figure 4.11 records the localization accuracy of SIFT as a performance behavior between Precision and Recall. This performance evaluates the global retrieval process by considering the data in the map as queries. Localization is tested by the given node queries, and the retrieval process excludes the querying data pattern from the matched outputs. Precision and Recall values are recorded starting from the first best node match and up to the n^{th} best match, where n is the total number of map patterns/images. The relationship indicates that 20% of the nearest neighbor images are 100% identified, and 60% of nearest neighbor images are identified with a precision of more than 90%. 100% Precision at 20% Recall means that the first nearest neighbor or top match is always classified to the correct node.

Figure 4.12 shows an environment place example subject to different values of Gaussian white noise with a standard deviation equal to 0.001, 0.01, 0.1. Figure 4.13 shows the effect of different noise values (in terms of standard deviation) on the Precision-Recall performance compared to the non-turbulent image data. It is clear from the curves that SIFT features and recognition are sensitive to intense noise interferences. Similarly, Figure 4.14 shows the effect of sensor resolution on retrieval performance. The default image size 640x480 is down-sampled to obtain four lower image resolutions of 352x288, 320x240, 176x144 and 160x120.

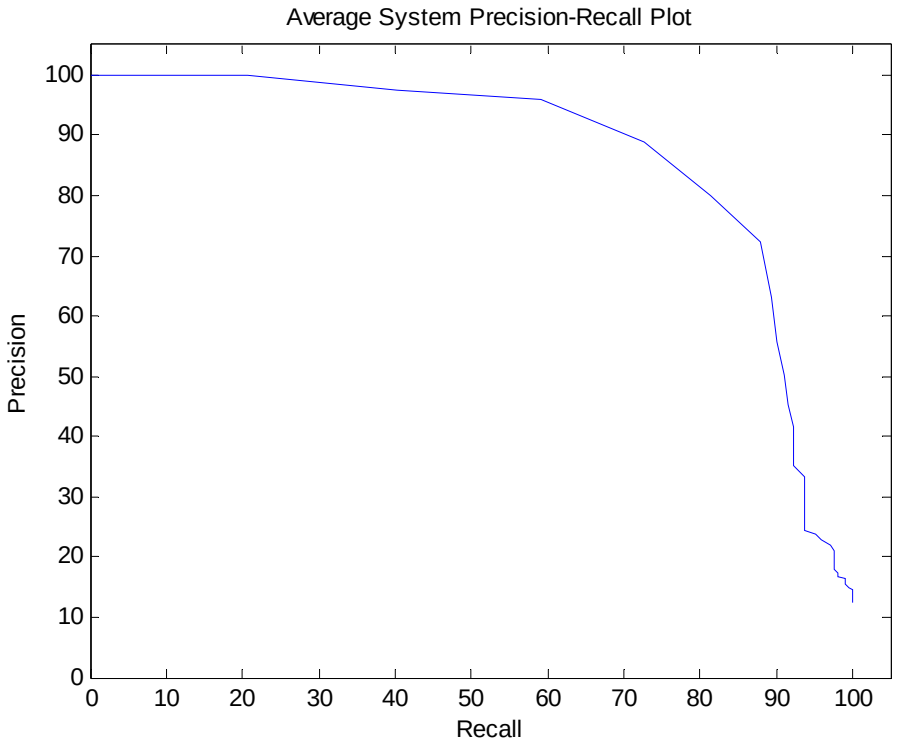


Figure 4.11. Precision-Recall performance for SIFT feature extraction.

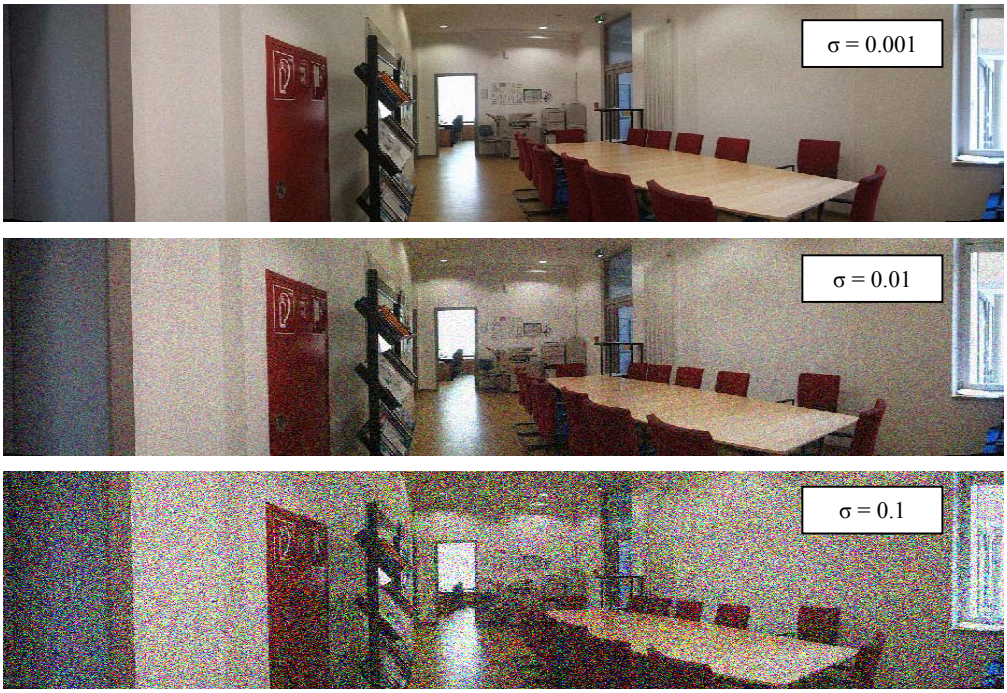


Figure 4.12. Image with different noise values.

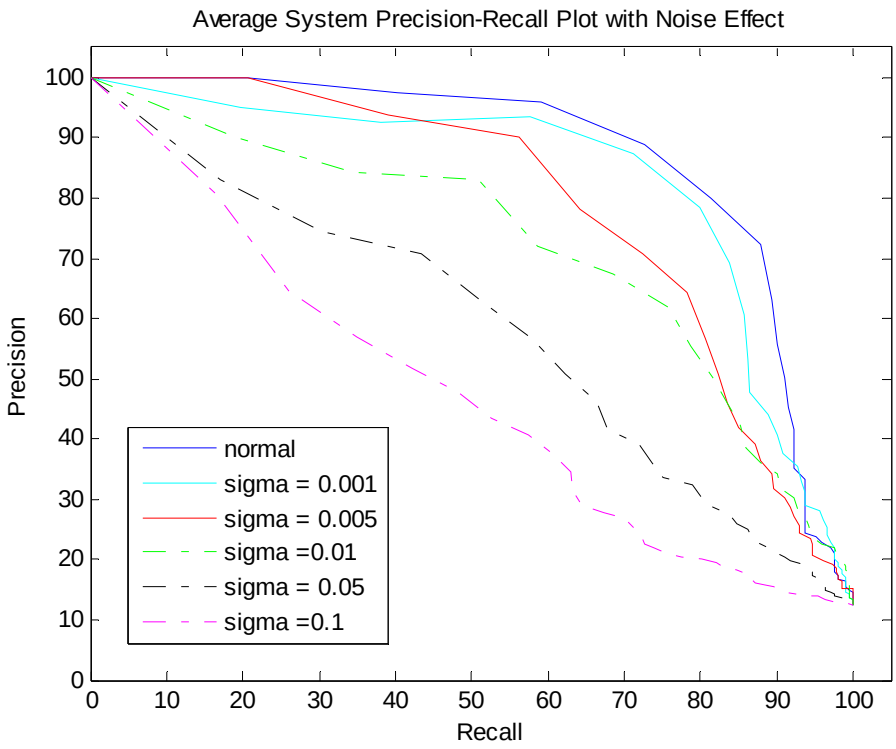


Figure 4.13. Retrieval Performance against Noise.

The results indicate that SIFT performance shows relatively good robustness to resolution. Figure 4.15 shows testing partial image matching, which can account for the scale. Partial image querying is correctly matched to similar views of other images, and vice versa. This indicates the robustness of local features to the scale, which is a significant advantage.

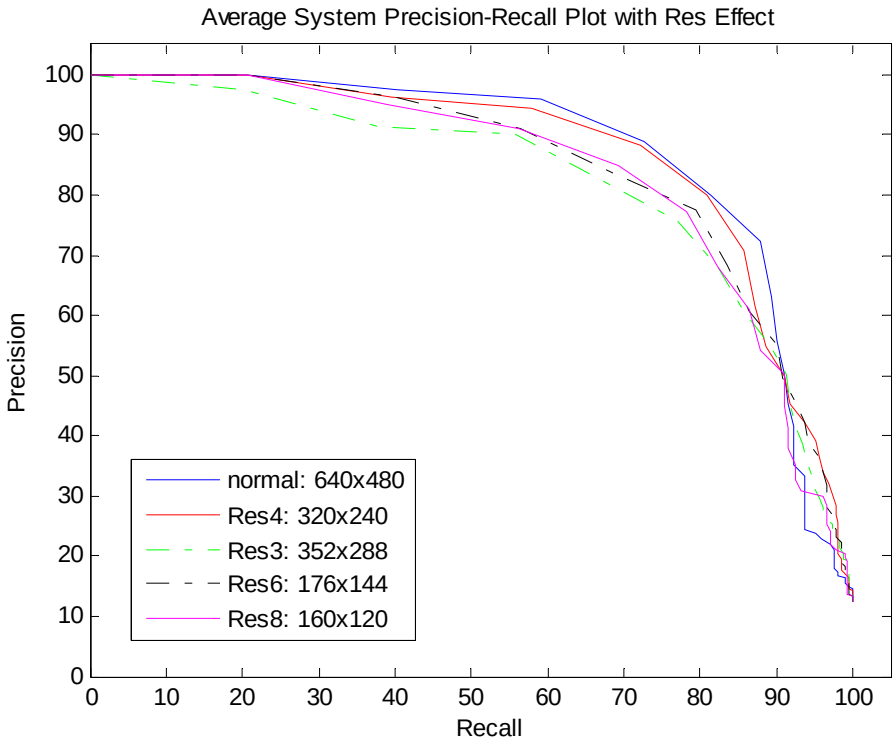


Figure 4.14. Retrieval Performance against Resolution.

Figure 4.15. SIFT robustness against scale.

4.9.2 Parameter Identification of the Accuracy Performance Measure

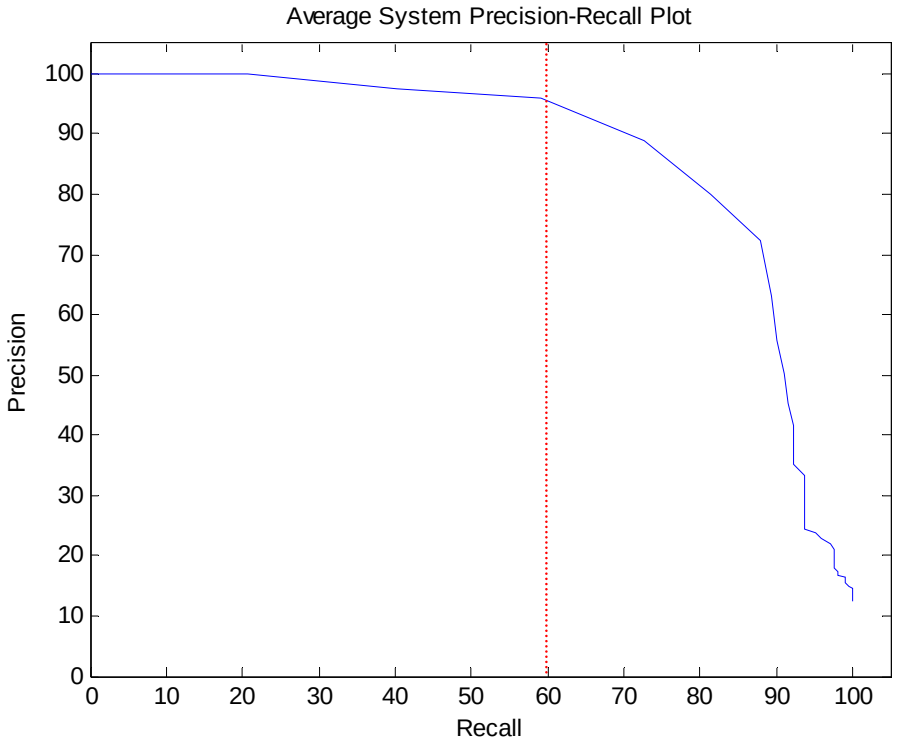
For specifying an exact measure for localization accuracy, a cost function E has been defined by (4.16). The function should be minimized to deduce the parameter β that indicates the ratio of Precision versus Recall. Localization accuracy is measured by the Precision at a fixed value of Recall which is identified by this cost function.

The value of E is optimized through experimental statistical measurements, where the SIFT-map previously constructed in 4.9.1 is utilized. In this experiment, the system is also tested by querying the data in the map, excluding the query from the results, and recording the equivalent Precision and Recall values starting from the first best match and up till the n^{th} best match, for the total number of map images n.

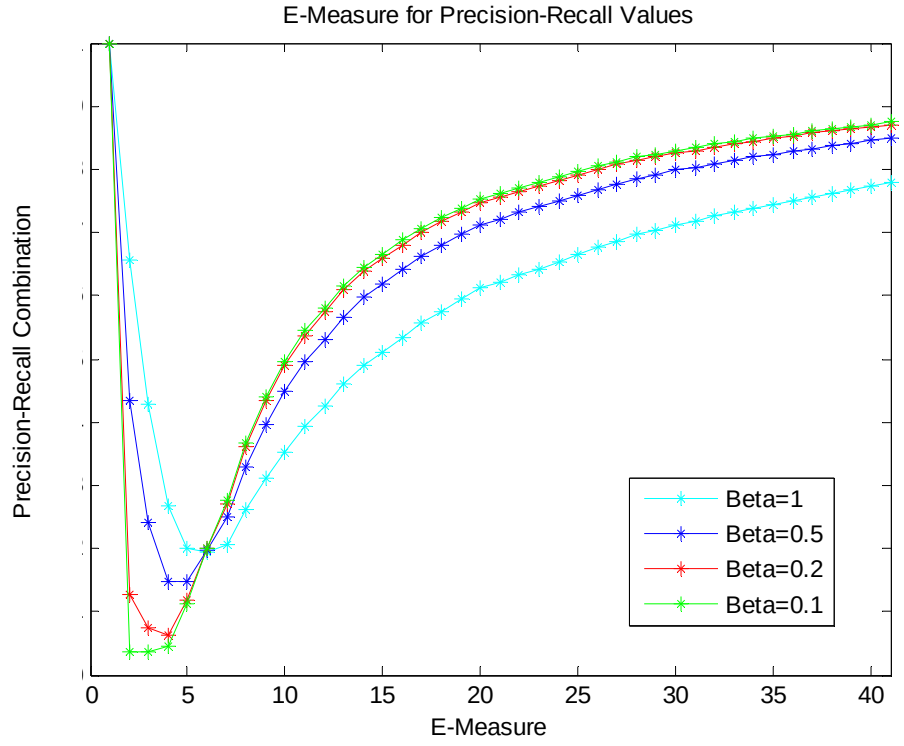
Figure 4.16-a shows the previously illustrated relation between Precision and Recall performance for SIFT retrieval process. Those Precision/Recall combinations are used to find the minimum E that solves for parameter β. Figure 4.16-b shows the E-Measure calculated using (4.16) versus the given Precision/Recall combinations, and for different values of $\beta = 1$, 0.5, 0.2, 0.1. Precision-Recall Combination values are recorded at the corresponding minimum values of E with for the different β values. Table 4.3 summarizes these combination values obtained from figure 4.16-b. The minimum E value occurs at $\beta = 0.1$, indicating the recommendation to set the localization accuracy measure as the Precision value at 20% Recall. Taking approximation, this means the first best match.

Nevertheless, we prefer to introduce some uncertainty or noise in the adopted localization accuracy measure. Therefore, β will be set to a slightly higher value equal to 0.5. This setting adds extra value for Recall and sets it to be as important as the Precision. The setting which regards additional significance of Recall implies testing localization more strictly. The increased recalled retrievals will cause a drop in the Precision value but on the other hand will create a distribution over the identified location. Moreover, the ability of recognition to retrieve successive correct matches will be of concern to evaluate, not just the ability to retrieve a single best match which is always a default used by other works.

Setting β to 0.5 means enforcing the condition that more than half of the correct patterns (60%) of a map node should be recognized when querying this node. This percentage



(a)



(b)

Figure 4.16. (a) Precision-Recall performance for SIFT feature extraction. (b) E- Measure for the Precision/Recall combinations.

is fair enough to judge the system as a localization module and as a general image retrieval system at the same time. In the next investigations, we will apply the case that the measure of the topological localization accuracy is the Precision value at 60% Recall.

Table 4.3. Performance measures versus parameter Beta

β	1	0.5	0.2	0.1
Precision (%)	80	96	97	100
Recall (%)	80	60	40	20

4.9.3 Clustering-Based Outlier Elimination

K-means clustering with a Cosine distance is applied on the training dataset for outlier detection. The unsupervised approach is suitable for the problem as discussed, because it is hard to obtain prior knowledge about classification labels for the SIFT keypoints, being fraud or real, weak or robust. The procedure starts by clustering keypoints associated with patterns that belong to the same topological node using *k*-means. The value of *k* is assigned to the least number of keypoints occurring in the patterns of the same node. Hereafter, clusters having 1-7 keypoints are removed.

Figure 4.17 shows the Precision versus Recall plot when removing 10%, 13% and 14% of the features as outliers, compared to the original features extracted by SIFT (normal). As the graph indicates, outlier removal has a small negative impact on the Precision. This is clearer with 14% outliers. Such a result is expected since outliers are distinguishable as an external fraud effect, but not as a characteristic of the scene. Though further SIFT keypoints elimination can have a negligible effect on the retrieval performance, eliminating only 13% of the keypoints as outliers is chosen, in order to preserve the same Precision that is obtained by the original SIFT feature extraction at 60% Recall.

4.9.4 Clustering-Based Feature Pruning

As previously mentioned, the vision feature space is in most cases of high dimension. For example, the main drawback of the SIFT features is that a huge number of keypoints is generated. This large size induces a negative influence on the general performance regarding accuracy and complexity. Therefore, in this case, feature pruning is highly recommended. Features have been pruned using clustering. In the literature, we outlined a clustering-based

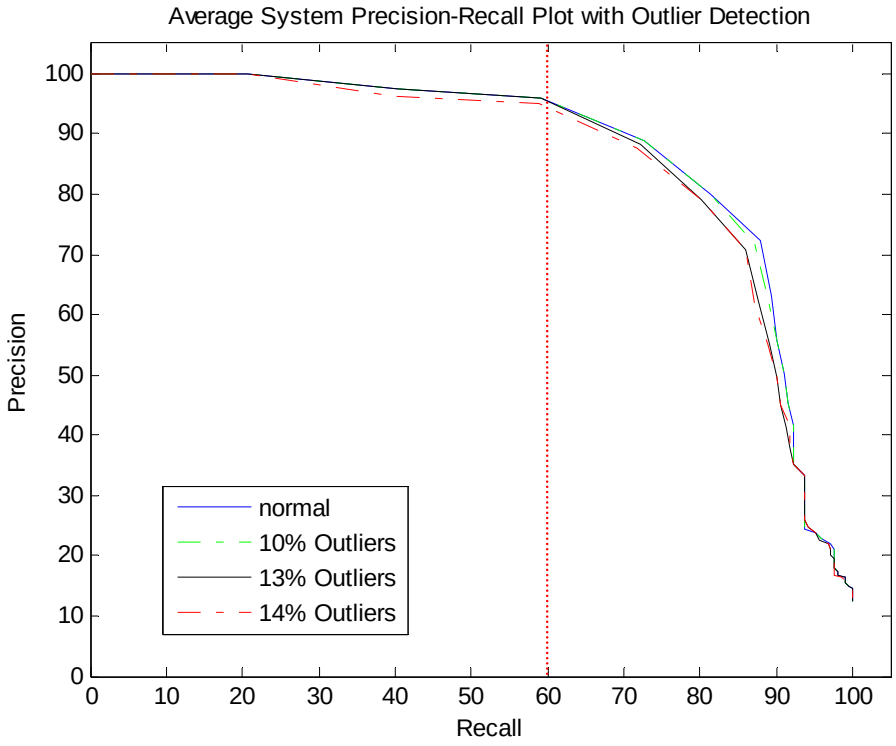


Figure 4.17. Retrieval performance after outlier removal.

approach for pruning SIFT features in an indoor environment [Ayers and Boutell, 2007]. The approach breaks SIFT keypoints of each place into a certain number of clusters using the *k*-means algorithm, and represents each class by mean values of the clusters. For matching, a nearest mean classifier is used for every SIFT keypoint in a test image to vote for the place class. The idea seems plausible to reduce the large size of keypoints. Excluding our proposed evaluation step for the features, both the authors' and our implementation appear quite similar.

We applied the authors' approach to Heidelberg's University feature-map after outliers were detected and eliminated. The number of tested clusters *k* is varied between 300 and 800. Figure 4.18 shows the performance results of the approach. The graph indicates non-satisfactory results in comparison to the original features performance. In addition, the first matches exhibit obvious low precision values, which means that the localization cannot recognize the first match efficiently (relatively high false positives). Localization recorded an average accuracy between 78 ~ 83 % precision at 60% Recall. The approach is reported, as coinciding with the results obtained in [Ayers and Boutell, 2007], as unsuccessful.

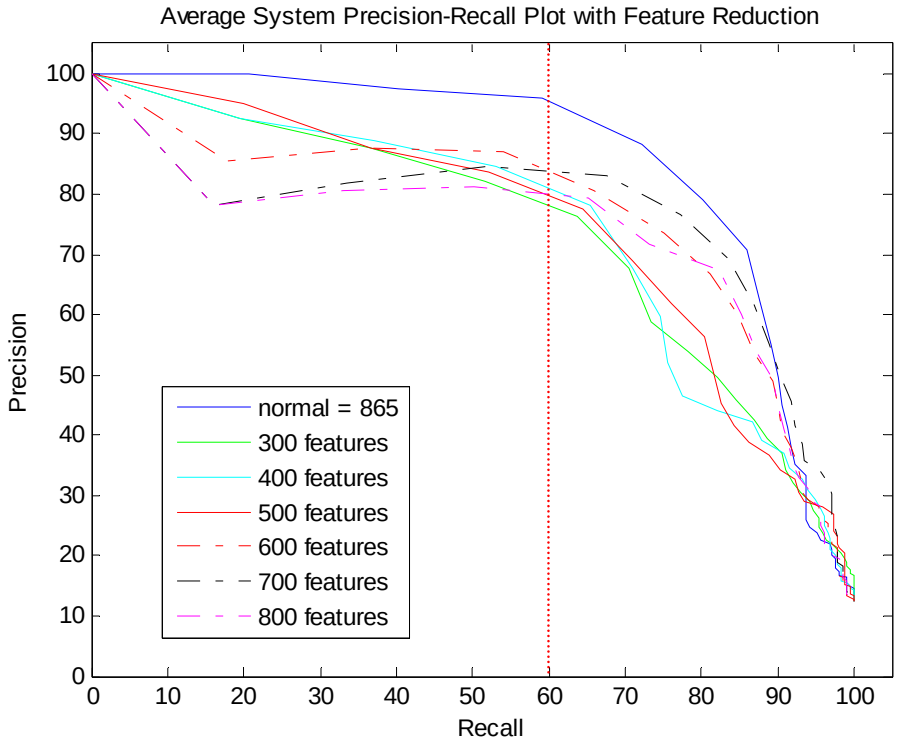


Figure 4.18. Performance of pruned features using *k*-means based on data redundancy.

4.9.5 Information-Theoretic Feature Evaluation

SIFT shows high efficiency for place recognition, as performance in figure 4.11 indicates. However, the features represent a great overhead on the localization. Though outlier removal reduced the size of the map by 13%, the overhead is still high. The clustering-based feature compression experimented in the previous section which performs on the native extracted features didn't prove its efficiency. The proposed information-theoretic evaluation is expected to solve those problems. Two datasets that form HEID are used for evaluating the localization: the first is the training dataset and used for the information-theoretic evaluation. The second is another testing dataset collected online from the environment, and is used to evaluate the localization using the filtered *entropy-based features* by the information-theoretic evaluation component. To localize the robot, the robot visits a node, executes sequential rotation actions to build the panoramic view of the place and then features extracted at the querying place are compared to the map. The Cosine distance defined by equation 4.2 is used in feature-to-feature matching with a threshold value of *thr* equal to 0.6. Matched image patterns for the topological place(s) are determined through feature majority votes.

The feature evaluation approach is applied on the training dataset after outliers are eliminated. The *entropy-based feature set* is generated as explained in section 4.6.4. We experimented with different values for the parameter Ψ (number of *feature categories*) in equation (4.9), between 10 and 1000 for more than 35000 keypoints in the dataset, and monitored the effect of eliminating keypoints of relatively high entropy values.

Figure 4.19 shows the localization accuracy versus different percentages of elimination of high-entropy keypoints in the training dataset, for different Ψ=10, 100 800 and 3936. The plot indicates that high cluster variation indicated by Ψ shows less accuracy (Ψ=3936). This identifies that the data undergo extra division and lose meaningful information content. For less cluster variation, the plots maintain an almost constant performance before it starts decreasing at 64% elimination. The plot still maintains high performance up to 72% for Ψ=800. The best performance in this study is obtained at 800 clusters. The graph shows that eliminating up to 64% of the keypoints as high-entropy keypoints has insignificant effect on the Precision. This means 64% of the keypoints do not share efficiently in the candidate codeword or place classification, but are rather an overhead on the localization. The figure

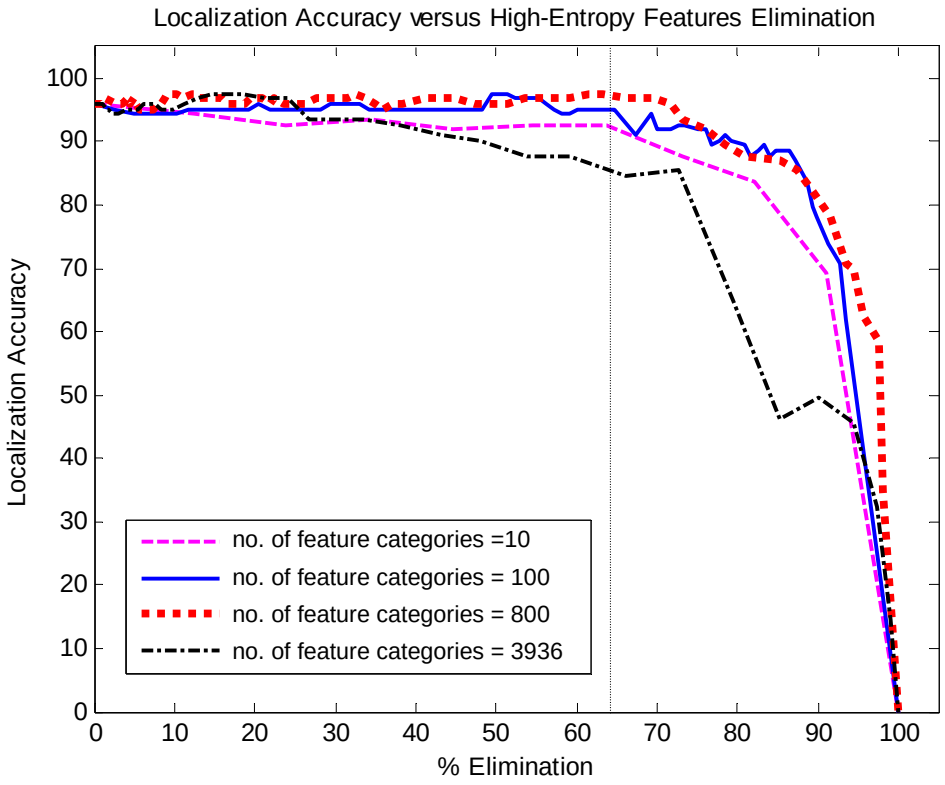


Figure 4.19. Average localization accuracy versus high-entropy features elimination.

also shows that eliminating up to 72% as high-entropy features still preserves the same precision for 800 clusters.

Figure 4.20 shows an environment example image with the keypoints extracted by SIFT (a), and after eliminating 44% of the keypoints as high-entropy features (b). The second image preserves only 511 keypoints, in comparison to 913 keypoints extracted by the normal SIFT, with 100% recognition precision for this place category.

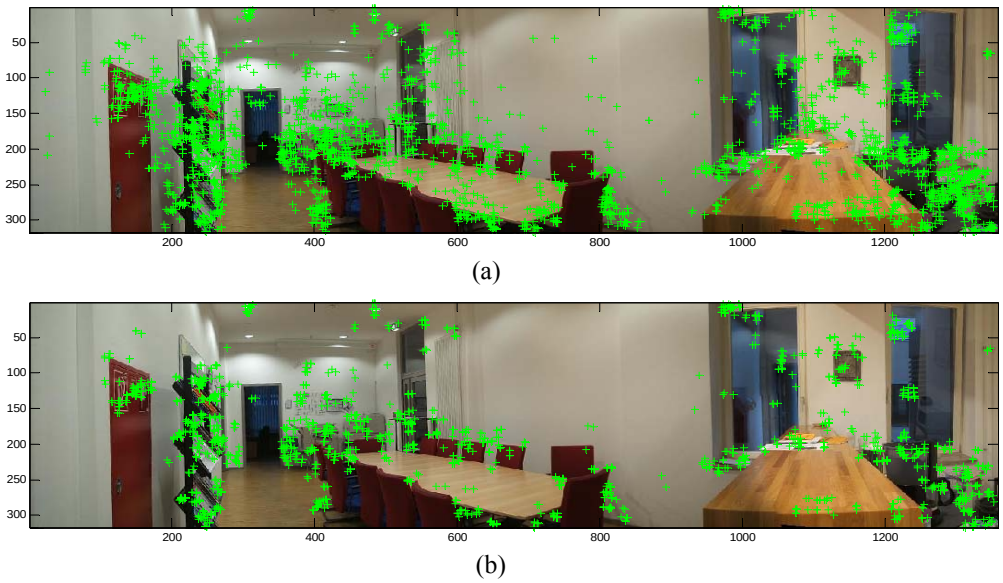
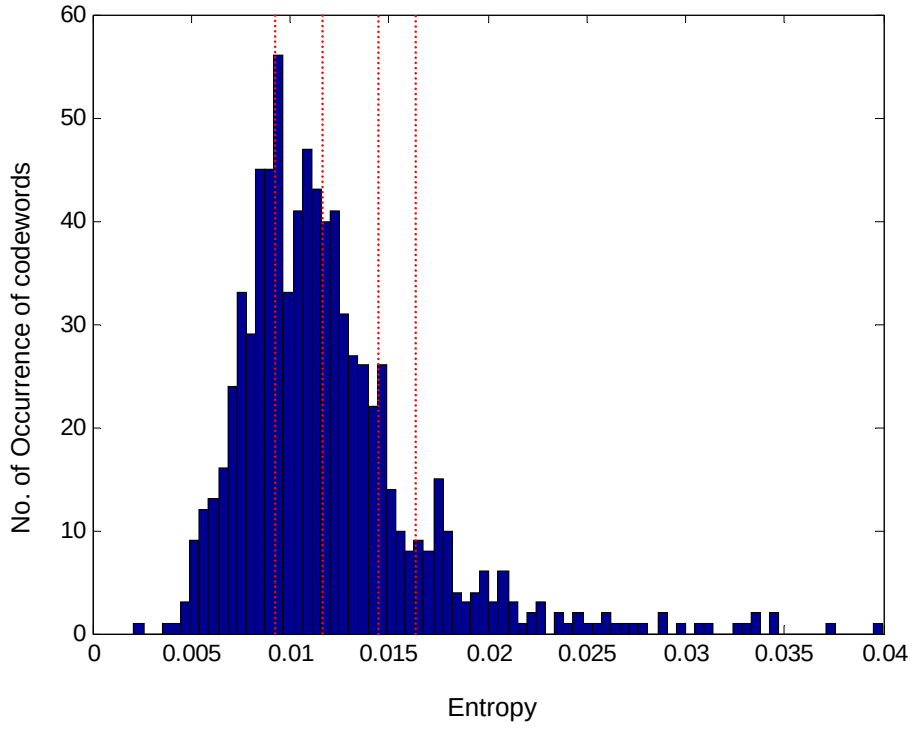
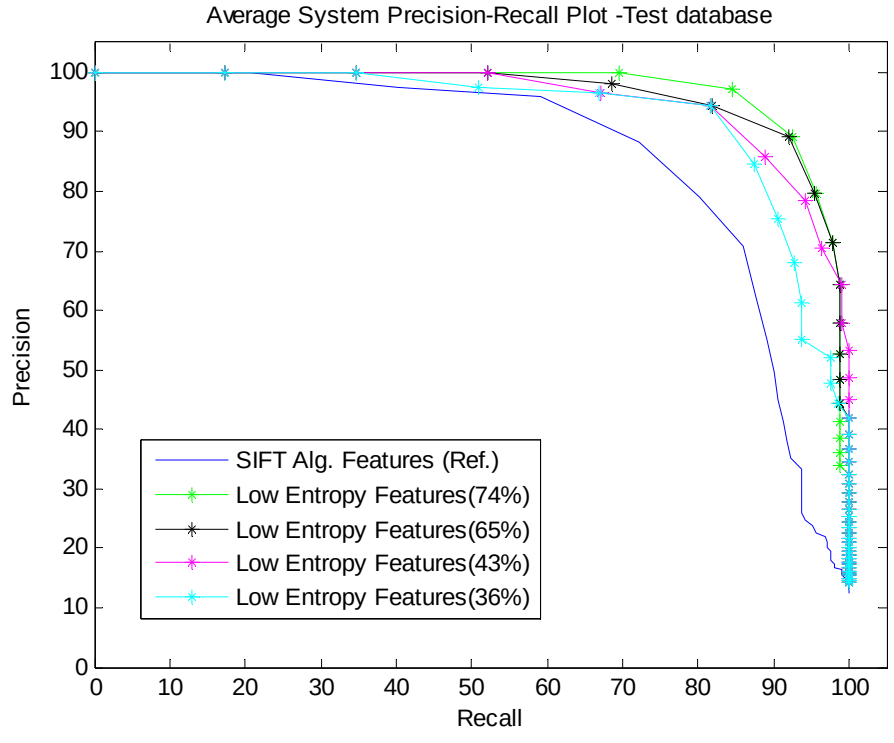


Figure 4.20. (a) An Indoor panoramic image with keypoints extracted by SIFT algorithm (No. of keypoints = 913). (b) Low-entropy keypoints after removal of high-entropy features (No. of keypoints =511).

Figure 4.21-a shows a histogram for the entropy of keypoints obtained at 800 cluster quantization. Different margins indicate thresholds for filtering low-entropy features with percentages (74%, 65%, 43%, and 36%). Figure 4.21-b shows the relation between Precision and Recall performance for the retrieval process using the second test dataset. As mentioned previously, the localization accuracy is set to be the Precision value at 60% Recall. The figure shows the retrieval performance of original SIFT features (Ref.) and the previous low-entropy features percentages. The four curves show much better performances than the one recorded by the original features extracted by SIFT. Eliminating 26% of the keypoints as high-entropy features (74% low-entropy features) achieves 100% Precision up till 70% Recall. This means the system recognizes more than four successive absolutely correct images out of an average of six that reside in the dataset. Moreover, eliminating up till 64%



(a)



(b)

Figure 4.21. (a) Entropy histogram for 800 keypoint clusters. (b) Average Precision-Recall performance for the test dataset by preserving low-entropy features and discarding high-entropy features. The percentage indicates the preserved low-entropy features referenced to original SIFT features.

(36% low-entropy features) achieves full precision at 34% Recall, and still achieves slightly better performance than the original SIFT at 60% Recall.

Consequently, it is concluded that almost 64% of the SIFT generated keypoints do not share efficiently in the topological node categorization, and hence are not informative features. For the different experimentations, the entropy-based feature map outperformed the initial SIFT-map, with the best performance obtained at 800 clusters (review figure 4.19).

The same range of the parameter Ψ is also tested using the Silhouette coefficient defined by (4.12). Average values of the coefficient, $\overline{S}$, have shown a monotonic increase over a small range [0.1508, 0.1823], which indicates accepted clustering quality and flexibility of the values of *k*. Consequently, any value for *k* in its studied spanned domain can be picked up flexibly without complications or severe performance degradation. The studied feature domain fortunately covers a relatively wide range due to the large size of vision data. This is possibly not the same when extracted features are of smaller size.

Table 4.4 illustrates the retrieval performance for some entropy-based feature maps which are employed in parts of the conducted studies. These are:

A. Feature map extracted by SIFT Algorithm
B. Feature map after outliers removal (13% Outliers)
C. Entropy-based feature map (74%), 26% removal of high-entropy features, training data B
D. Entropy-based feature map (65%), 35% removal of high-entropy features, training data B
E. Entropy-based feature map (43%), 57% removal of high-entropy features, training data B

Results indicate that the same high recognition/localization accuracy using SIFT (A) can be obtained through the low-entropy feature set, which has the impact of reducing almost 2/3 of the overhead (E). The average retrieval precision is 96%, which implies a high confidence that the localization is correct.

Table 4.4. Retrieval performance: Performance index versus feature-map

Performance index/Feature-map	**A**	**B**	**C**	**D**	**E**
Average KPs (features)per image	1000	860	635	560	370
Precision at 60% Recall (%)	**95.9**	95.9	100	99.2	**96.7**
Search and Retrieval Time (sec)	19.8	16.5	12.	10.2	6.9
Space reduction (%)	-	14	36.5	44	**63**
Time reduction (%)	-	16.7	38.4	48.5	**65.2**

4.9.6 Codebook Performance

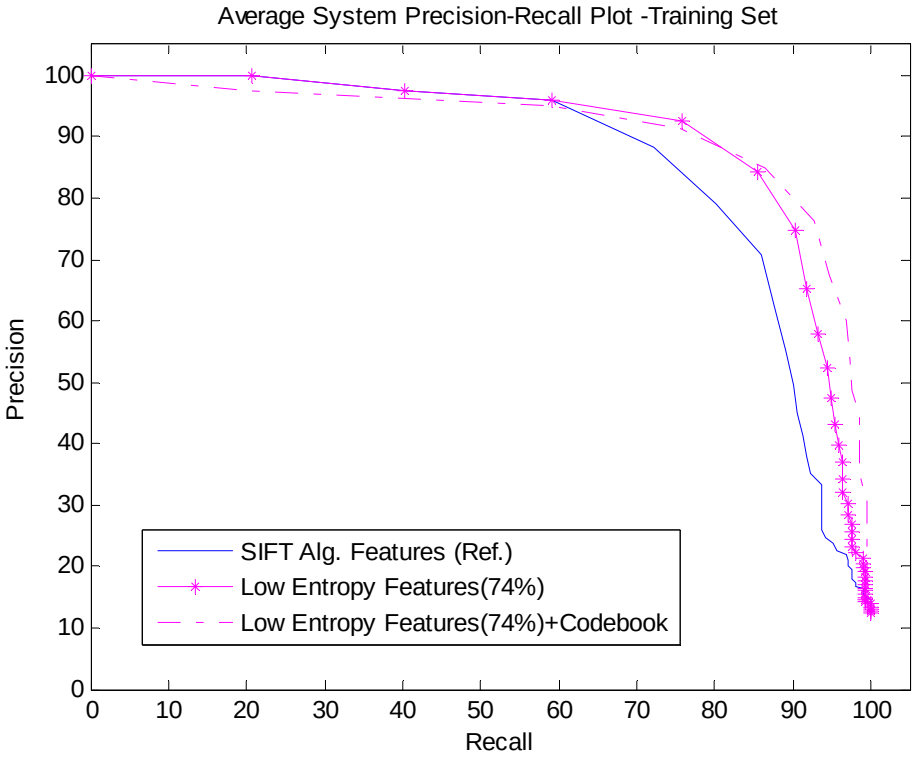
In the codebook matching, extracted keypoints from testing queries are compared against the codewords of the CB. The Cosine distance is used for keypoint-to-codeword matching, with a threshold value for *thr* equal to 0.5. Far distanced matches are discarded and the matched place is determined through the majority votes acquired. Since the CB preserves only one value on behalf of a keypoint cluster, the size of the preserved features is again much more reduced than the low-entropy features obtained in the previous section.

The results of two CB examples are shown: the first CB example is obtained from the low-entropy feature set reduced to 74% of the original features, and has 3845 entries. The second CB example is based on a reduced feature set to 36%, and has 2739 entries.

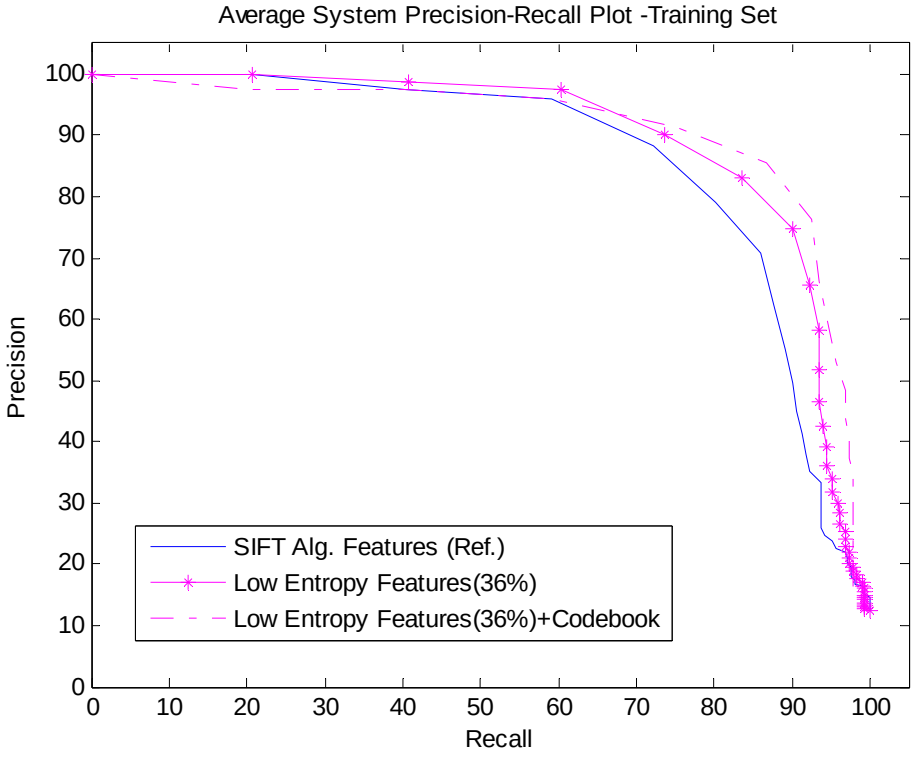
Figure 4.22 shows the CB performance in comparison to the initial SIFT-map and the entropy-based features of the training dataset. Localization using the CB has approximately a similar performance to the entropy-based features. Localization performance still preserves almost the same precision for the two used CBs (94-96%), in comparison to the original SIFT features (96%). The CB, however, provides significant reduction for the data stored compared to other maps. This reduction is about 90% in comparison to original SIFT features, and about 80% in comparison to the entropy-based features (hint: SIFT-map ~35K total keypoints; 74% low-entropy features map ~26K total keypoints; 36% low-entropy features map ~13K total keypoints). The CB results in a similar linear reduction in the matching time by almost 90% as well.

Figure 4.23 shows the performance for the two CB examples compared to the initial SIFT-map, when using the test dataset. The two CBs are approximately similar in performance as the original SIFT. The second CB (based on only 36% of low-entropy features) achieves 100% Precision for the top 1-2 place matches. Hence, it is concluded that a feature CB is worthwhile to gain high localization performance, which is similar to the original feature extraction performance. It saves, however, much of the overhead required for the storage of localization map and the pairwise matching of the data.

A querying example for an office scene in the tested environment, HEID, is shown in Figure 4.24. The figure shows the top 4 retrievals by the place matching component. The retrievals are correctly identified by the 36% low-entropy map and the Codebook based on



(a)



(b)

Figure 4.22. Average retrieval and localization performance for the training dataset using CB based on (a) 26% reduction of SIFT keypoints (3845 Entry CB) (b) 64% reduction of SIFT keypoints (2739 Entry CB).

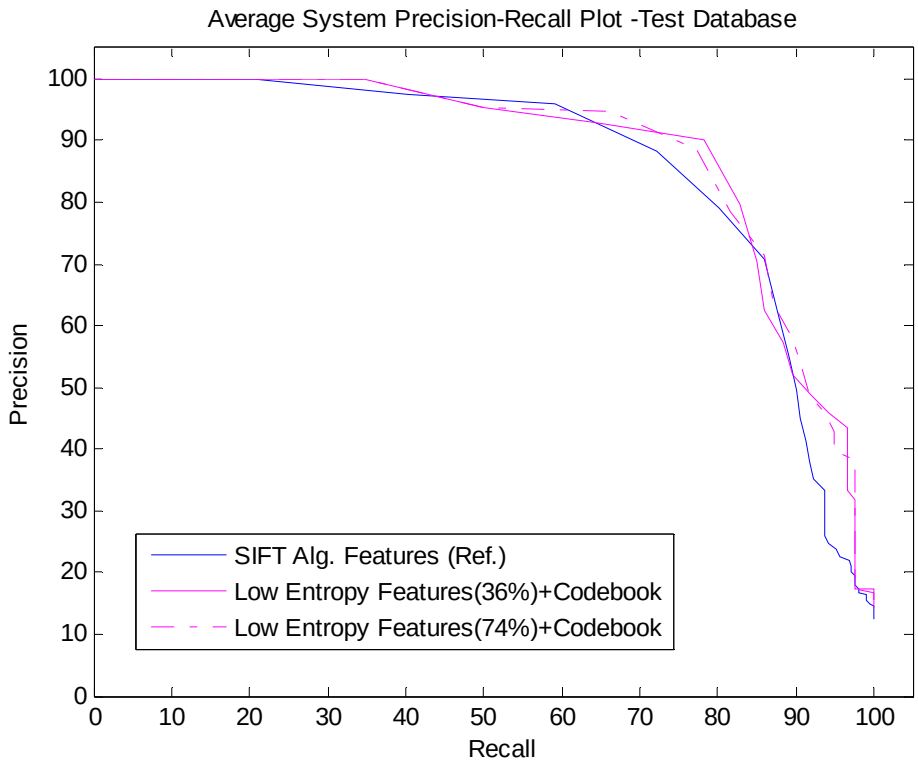


Figure 4.23. Average retrieval and localization performance for the test dataset using two different CBs.

the same low-entropy feature set. The figure highlights the number of features extracted for the query by the original SIFT algorithm, and the number of low-entropy features and codewords stored for every retrieved image in the corresponding map. Scene dynamics and illuminations variation are clearly obvious in the map image examples.

Table 4.5 provides a summary for the gains attained by the proposed information-theoretic map building and localization approach. The data size and retrieval performance for some of the proposed reduced feature maps, which are employed in parts of the conducted studies, are given. In the table, time reduction is not equal to space reduction since this time concerns the localization process that includes additional feature extraction time of the query. The different proposed maps indicate that localization accuracy can be adapted efficiently to the available resources, and in other words, performance parameters (accuracy, complexity) are simultaneously tuned with the need. The table summarizes the following studies:

A. Feature map extracted by SIFT Algorithm
B. Feature map after outlier removal (~13% Outliers)

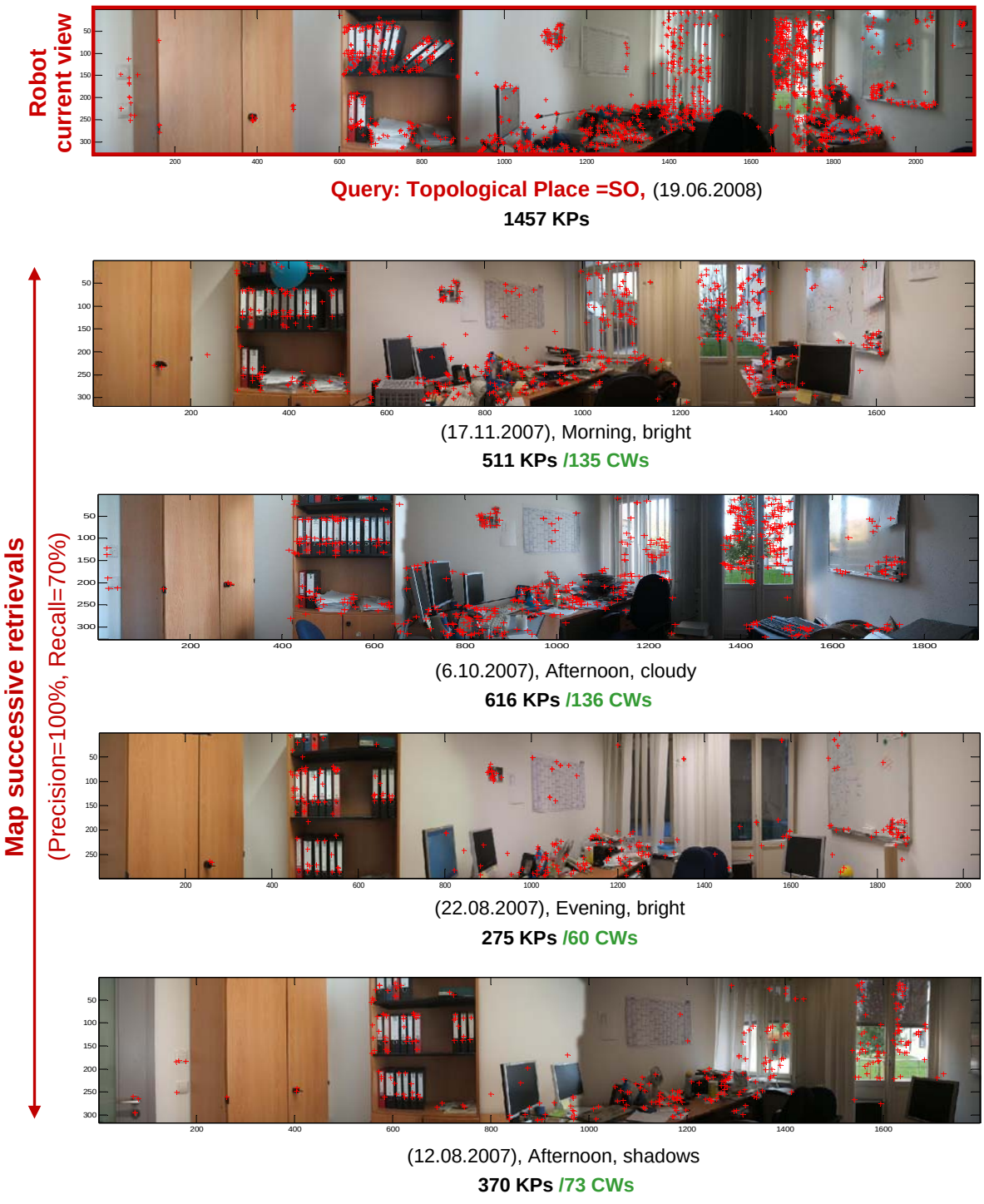


Figure 4.24. Illustrative localization example: A wide-view image for an office environment. Top image is the query image. The following images are the top 4 place matches, all correctly classified. Each image shows the associated keypoints extracted for the query and those acquired as reduced low-entropy features and codewords versions, which are stored in the topological map. Additionally, each image illustrates a different weather and illumination acquisition conditions. Scene dynamics is also obvious through new, moved or missing objects.

C. Low-Entropy feature map (74 %), training data B

D. Low-Entropy feature map (36 %), training data B

E. Codebook based on 36% low-entropy feature set

Table 4.5. Data size and retrieval performance: Performance index versus feature-map

Performance index/Feature-map	**A**	**B**	**C**	**D**	**E**
Average KPs per Image	1000	868	640	320	67
Precision at 60% Recall (%)	**95.5**	95.5	100	**96.9**	**93.6**
Localization Time[2] *(sec)*	22.81	19.61	17.72	10.36	4.96
Map Size(MBytes)	40.07	34.8	25.65	12.81	2.67
Space reduction (%)	-	13.2	36	**68**	**93.3**
Time reduction (%)	-	14	22.3	**54.6**	**78.3**

4.9.7 Localization Performance under Acquisition Disturbances

Figure 4.25 shows image examples for the meeting place with disturbed acquisition conditions. The images have been acquired in two situations. In the first situation, the robot platform is driven with relatively high velocity, introducing high vibrations in the camera sensor. In the second situation, the robot platform is driven with lower velocity yielding less camera vibrations. These situations, accordingly, influence the quality of image stitching. It is necessary to see the effect of those disturbances and the consecutive stitching quality on the localization performance. Two webcams with two different resolutions have been used as well: Logitech 4000 pro (320x240) and Logitech C600 (640x480). This is because the image resolution is expected to have a consecutive effect on the image stitching quality as well.

The introduced high vibrations lead to bad-quality stitching as illustrated by figures 4.25-a,c. Figures (a) and (b) show the panoramic view constructed by the low resolution camera from 3-image stitching procedure, while figures (c) and (d) show the view constructed by the high resolution camera from 6-image stitching procedure. The images show that higher resolution images undergo low-quality stitching when the camera exhibits severe vibrations.

Localization performance under the disturbed acquisition conditions is summarized in table 4.6. Employing a low-resolution sensor provides more robust localization towards the low-quality stitching caused by severe vibrations. The CB performance is slightly affected by

[2] Localization time includes 3.11 seconds, which is the SIFT execution time on the specified machine.

(a)

(b)

(c)

(d)

Figure 4.25. Examples for disturbed acquisition conditions. Stitching quality is influenced by sensor vibration and resolution: (a) 3-image stitching (320x240), high vibrations. (b) 3-image stitching (320x240), low vibrations. (c) 6-image stitching (640x480), high vibrations. (b) 6-image stitching (640x480), low vibrations.

the stitching quality than the entropy-based features. On the other hand, employing a high-resolution sensor provides less robust localization in a similar case. However, employing a high-resolution sensor shows good performance when stitching is of high-quality, which can be induced through a stabilized platform. Stabilization is guaranteed by acquiring the image shots after the robot executes a rotation control then pauses, instead of acquiring the shots while the robot platform is in motion.

Table 4.7 shows the processed time needed for stitching several images (2, 3, 4 and 6 images) by the two cameras). Stitching with the low-resolution sensor requires almost one second for every single stitching. This time is four times doubled with the high-resolution

sensor since the feature extraction generates larger number of features. Experimentations have demonstrated stitching up till 3 images are enough for capturing the information of the scene.

Table 4.6. Localization performance with disturbed acquisition conditions

	Logitech 4000 pro (320x240)			
	Less vibrations/ Good-quality stitching		*Severe vibrations/ Low-quality stitching*	
Parameter / **Method**	**Entropy-based features**	**Codebook**	**Entropy-based features**	**Codebook**
Avg. matching time (sec)	2.8475	0.8532	2.1094	0.5726
Avg. localization time (sec)	4.009	2.9060	4.1622	2.6254
Best match localization (%)	100	50	100	100
Distribution localization[3] *(%)*	100	50	100	50

	Logitech C600 (640x480)			
	Less vibrations/ Good-quality stitching		*Severe vibrations/ Low-quality stitching*	
Parameter / **Method**	**Entropy-based features**	**Codebook**	**Entropy-based features**	**Codebook**
Avg. matching time (sec)	9.8620	2.5056	8.6820	1.9469
Avg. localization time (sec)	17.9329	10.5709	16.6734	10.0156
Best match localization (%)	100	100	25	25
Distribution localization[3] *(%)*	100	75	75	25

Table 4.7. Stitching performance. Stitching time versus resolution and number of images

	Stitching time(sec)			
	2 images	*3 images*	*4 images*	*6 images*
Logitech 4000 pro (320x240)	1.0955	2.0528	2.9486	4.3467
Logitech C600 (640x480)	4.3019	8.0709	11.5317	16.3182

4.10 Summary

A real challenge for a mobile robot is to acquire user-independent abilities to interpret and represent the structure of the environment in a suitable and efficient way. Large and cluttered environments, together with the massive data provided by sensors, can induce a

[3] 60% Recall localization.

highly-complex environment model for navigation. These factors can negatively influence the robot's navigational performance, notably with the accuracy and computational complexity of localization. Therefore, it is necessary to fit the right and exact amount of information to that required by the localization task, and in its simplest form.

This chapter has introduced an information-theoretic solution approach for environment modeling, and applied it for topological map building and localization. The approach is based on selecting a set of relevant features that contribute to location uncertainty minimization, and storing the features in their simplest form. This is achieved by evaluating environment extracted features based on measured transinformation values. Higher transinformation values indicate distinguishness of features, providing capability to resolve location ambiguities and worth for employment in a map model. The proposed approach is flexible in not adhering to specific sensors or feature extraction methodologies.

A solution structure has been laid out, providing robust, computationally efficient, as well as accurate localization. In this structure, two feature forms are generated as a result of employing an entropy-based filtering criterion combined with a clustering technique; entropy-based features and codewords. The first form is a reduced relevant feature set, while the second is a compressed form of the first. An implementation for the solution structure, using vision-based local feature extraction, has proven efficiency of both generated maps (entropy-based feature map and codebook) for mobile robot topological localization. The localization performance exhibits higher accuracy at lower space and time complexities. The codeword map has induced significant localization computational performance (ten times faster with the same attained localization accuracy). The gains of the proposed modeling approach have been quantified, which shows its capability for adapting to the available resources efficiently.

The proposed solution provides answers to important environment modeling inquests, such as which natural features or landmarks constitute relevance for incorporating into a map in unstructured, cluttered and dynamic environments, and what is an efficient representation of these data to minimize computational complexity, an issue that is extremely important for systems working in real-time. The performance evaluation of the solution approach, especially for the accuracy, has proven that the stand-alone topological unit can be integrated into a hierarchical design to assist scalable and computationally efficient geometric localization. This is the extended work that will be introduced in chapter six.

Chapter 5

Evaluation Study on COLD Benchmarking Database

5.1 Introduction

In this chapter, the information-theoretic environment modeling approach proposed in chapter four is tested on a recently-constructed standard benchmark database (COLD), primarily targeted for robot topological localization and map building. The tested images are acquired using an omnidirectional vision sensor and are subject to varying conditions of illumination and scene dynamics. Since it is important, in the point of view of vision-based solutions, to ensure robustness of solution against sensor resolution and dynamic variations, the benchmark study is relevant for the evaluation and validation of the proposed structure and methods. The considered evaluation is summarized in [Rady et al., 2010a].

Moreover, the chapter draws a comparison for the performances attained by HEID and COLD datasets on one side. On another side, it outlines customization for the solution approach according to the type of environment and sensor. The efficiency of the information-theoretic solution, which is based on relevant information filtration to minimize uncertainty and computational complexity, is once again elaborated and explicitly discussed.

The chapter is structured as follows: First, the benchmarking database is described in section 2. The solution approach application, experimentation and results for the benchmarking database are presented secondly in section 3. Section 4 compares the results obtained with that of the previous chapter, and highlights differences and conclusions. The chapter closes with a short summary about the purpose of the chapter and the results obtained.

5.2 COLD Database Description

COLD (COsy Localization Database) [Pronobis and Caputo, 2009; URL:COLD] is a large-scale testing environment for evaluating vision-based localization systems aiming to work on mobile platforms in realistic settings. The database provides a large versatile set of image sequences acquired at three different laboratory environments in European cities (Saarbrücken, Freiburg and Ljubljana) under different natural variations and dynamics. Similar place categories or room functionalities are found in all of the three databases. These represent different office rooms, corridors, kitchen, etc. Table 5.1 lists the types of rooms used at each of the three laboratories.

Perspective and omnidirectional video sequences were recorded for the database using different mobile robot platforms. The three mobile platforms engaged in the data acquisition at the three laboratories are shown in figure 5.1; with the parameters and settings of the camera sensors attached to each robot platform in table 5.2. Laser range scans and odometry data were also captured for most of the sequences. In each laboratory, data were acquired in all rooms using the same camera setup. Data acquisition process was done under different short-term dynamic changes caused by weather and illumination conditions (cloudy, sunny, night) across a time span of 2-3 days. Other dynamic activities like people wandering around and missing or newly added objects were introduced. Unfortunately, the videos have been acquired with low quality cameras (see figures 5.4 and 5.5). This has been reported in another work as well [Pronobis, 2011].

Table 5.1. List of types of rooms used during image acquisition at each of the three labs

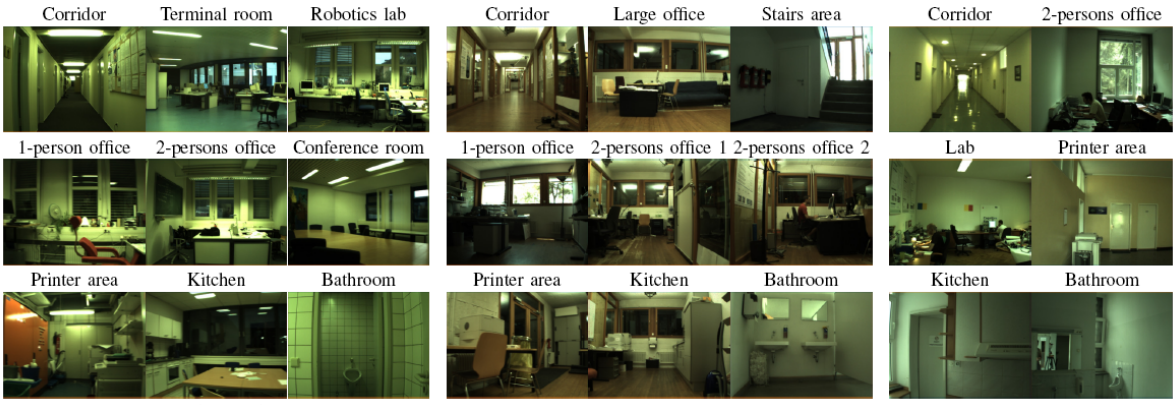
LAB	Corridor	Terminal room	Robotics lab	1-person office	2-persons office	Conference room	Printer area	Kitchen	Bath room	Large office	Stairs area	Lab
Saarb.	√	√	√	√	√	√	√	√	√	-	-	-
Freib.	√	-	-	√	√	-	√	√	√	√	√	-
Ljubl.	√	-	-	-	√	-	√	√	√	-	-	√

Table 5.2. Parameters and settings of the cameras for each robot platform

Robot platform	ActivMedia PeopleBot Saarbrücken		ActivMedia Pioneer-3 Freiburg		iRobot ATRV-Mini Ljubljana	
Camera type	Perspective	Omni	Perspective	Omni	Perspective	Omni
Frame rate	5 fps					
Resolution	640×480 pixels, Bayer color pattern					
Exposure	Automatic					
Field of view	68.9° × 54.4°	—	68.9° × 54.4°	—	68.9° × 54.4°	—
Height	140cm	116cm	66cm	91cm	159cm	153cm

Since the COLD database provides an ideal and flexible test bed for assessing robustness of localization and recognition solutions subject to illumination and dynamic changes, we use an exemplar part of this large database to validate the proposed information-theoretic approach in the next section. Figures 5.2 and 5.3 show some example images for the interiors of rooms at the three laboratory environments as reported by the database developers in their description documents. The figures show images captured using perspective and omni-directional cameras respectively.

Figure 5.1. The three mobile platforms employed for image acquisition at the laboratories [Pronobis and Caputo, 2009].



(a) Saarbrücken (b) Freiburg (c) Ljubljana

Figure 5.2. Example images from the three lab environments acquired by perspective camera showing the interiors of rooms [Ullah et al., 2008].

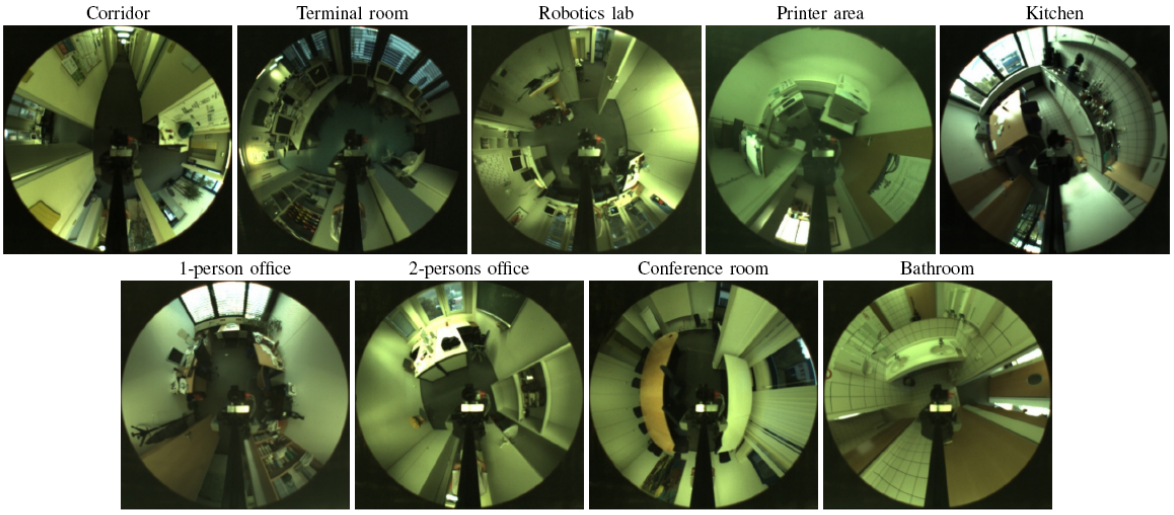


Figure 5.3. Examples of images of Saarbrücken database acquired by omnidirectional camera showing the interiors of rooms [Ullah et al., 2008].

5.3 Experimentation and Results

The COLD-Saarbrücken omnidirectional extended sequences (B) database is chosen in order to experiment the information-theoretic topological solution. The database contains omnidirectional image videos acquired by an omnidirectional sensor (see figure 5.3). In each video sequence, the robot navigates visiting five different functional areas in about 34 seconds (corridor, 1-person office, kitchen, bathroom and a printer area). Nine different places among those categories are selected as topological map nodes. The selected place images are shown in figure 5.4. Before applying feature extraction and generating an original SIFT feature map,

the omnidirectional images are unwrapped by projecting them on a cylinder surface to obtain normal view images (see figure 5.5). The resolution of omnidirectional image is 640x480 pixels, which accounted for a 1003x199 slightly distorted image after unwrapping.

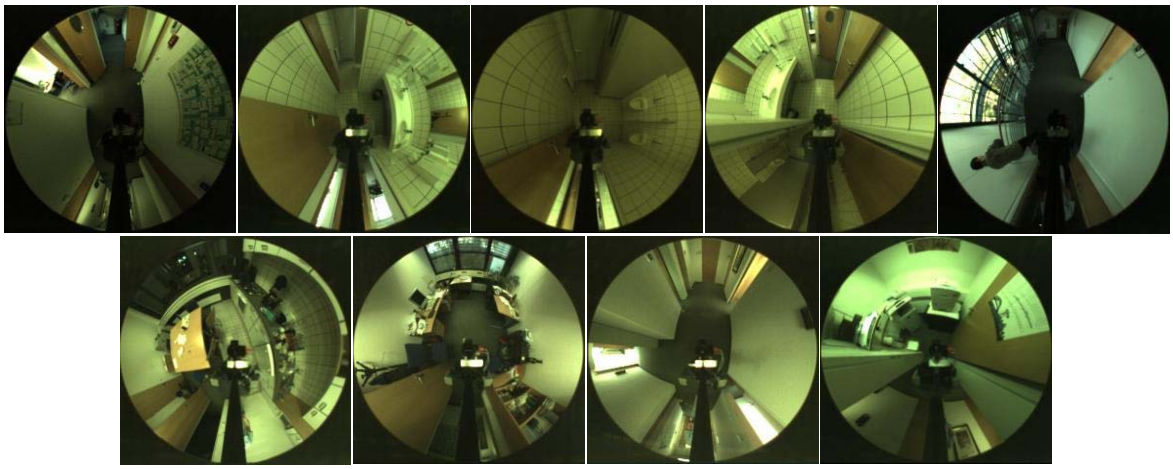


Figure 5.4. Nine-place example images for five functional categories in Saarbrücken-COLD (B) Database (corridor, bathroom, kitchen, office room & printer area).

Each node in the map is represented by 3-9 images, according to the existing amount of variation accounting for the lighting conditions, scene dynamics and robot pose. A dataset is prepared for feature evaluation with a total of 49 images and is constructed from three videos with the three different dynamic categories (cloudy, sunny and night). A second testing dataset is prepared for the performance evaluation and is constructed from six videos with another intensity variation which is indicated by the benchmark developers. The test dataset has 52 images. Figure 5.5 shows omnidirectional and unwrapped normal-view images of two different place examples of the first dataset. The images illustrate severe dynamic variations (darkness, shadows, viewpoint change) which influence the testing environment.

SIFT feature extraction is applied on data of the first dataset. The average number of keypoints extracted per image is 333, each with 128 dimensions. Assuming that a single number occupies 8 storage bytes, the size of original SIFT map becomes 15.92 Megabytes.

In localization, the Cosine distance is used for matching images, together with a threshold to discard far distanced keypoint matches (section 4.6.2). The threshold value is set to 0.6 in keypoint-to-keypoint matching for the entropy-based map, and to 0.5 in keypoint-

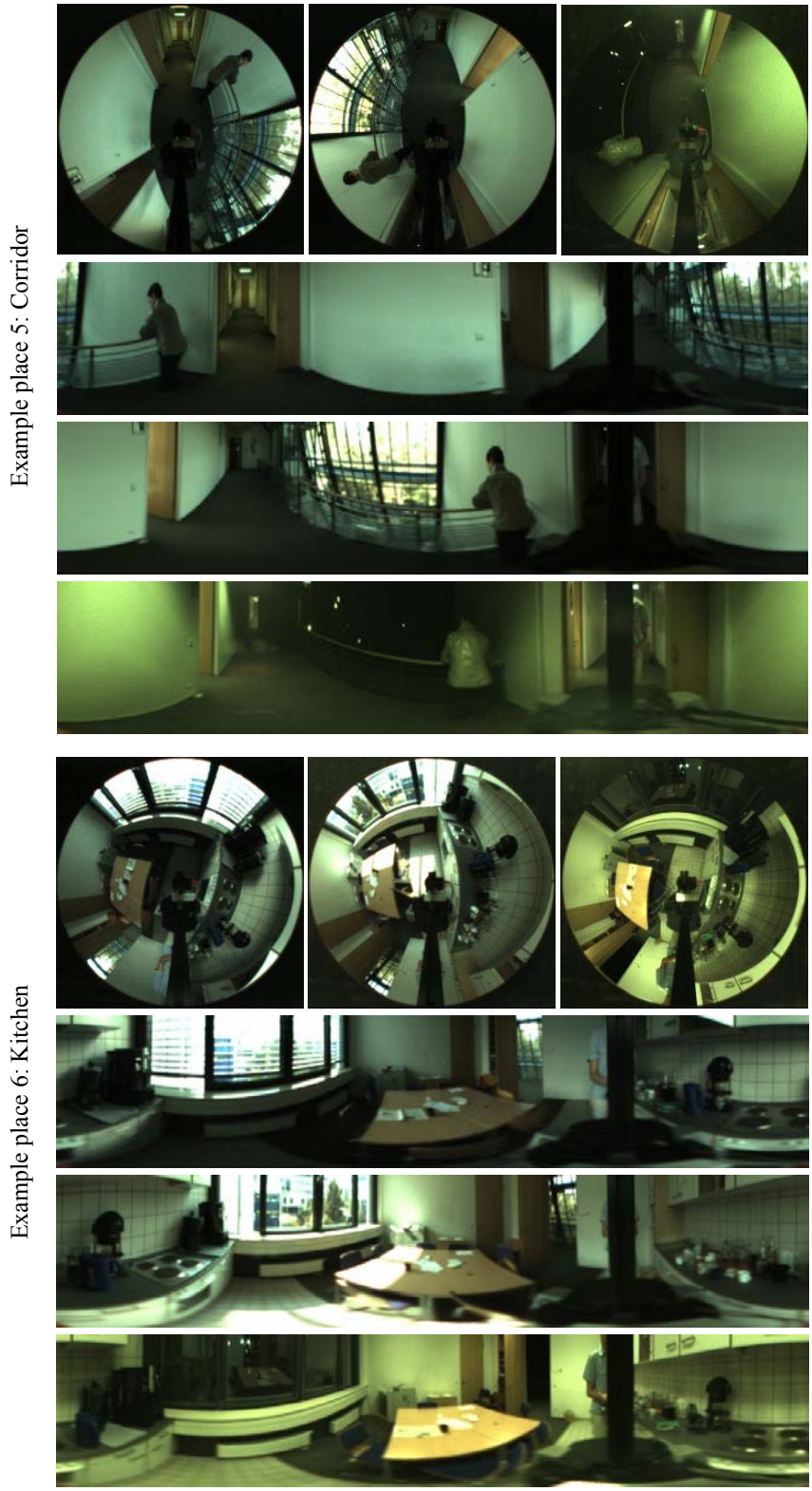


Figure 5.5. Examples from the training dataset for the different weather and illumination conditions (cloudy, sunny and night). The figure shows the omnidirectional images and their unwrapped versions. The images indicate severe dynamic changes with respect to scene, illumination and robot pose.

to-codeword matching using the CB. The identity of queried place is determined through the keypoint or codeword majority votes. Similarly, as with HEID, localization accuracy is set to the calculated Precision value at 60% Recall (i.e. the ratio of correctly identified images from the entire retrieved set, on condition that 60% of the correct images that reside in the map corpus are retrieved). In the first COLD dataset, localization using the original SIFT features recorded 80% accuracy as shown in figure 5.6. The figure shows the complete retrieval performance, in which the first 1-2 images (20% Recall) show high localization accuracy (over 90%) as measured by the Precision value. After 20% Recall, the accuracy drops.

In the *k*-means clustering of outlier detection, 9% of keypoints are eliminated after discarding small clusters (≤ 5). The value of k is set to be a function of both the number of images per place and the average number of extracted keypoints per image, in order to avoid discarding too many features from images of locations with few detected features. Filtered data after outlier elimination have almost identical Precision-Recall performance behavior as the original features extracted by SIFT. Figure 5.6 shows the performance of the original SIFT features and after outlier detection and elimination. Similarly as done in chapter 4, the filtered dataset will be used as the *training dataset* in the information-theoretic evaluation.

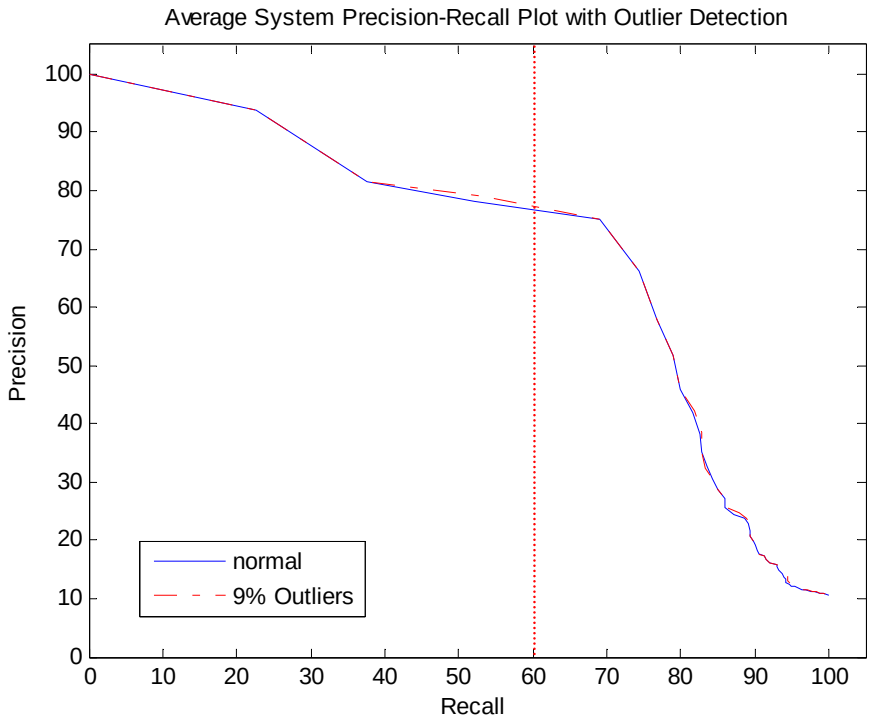


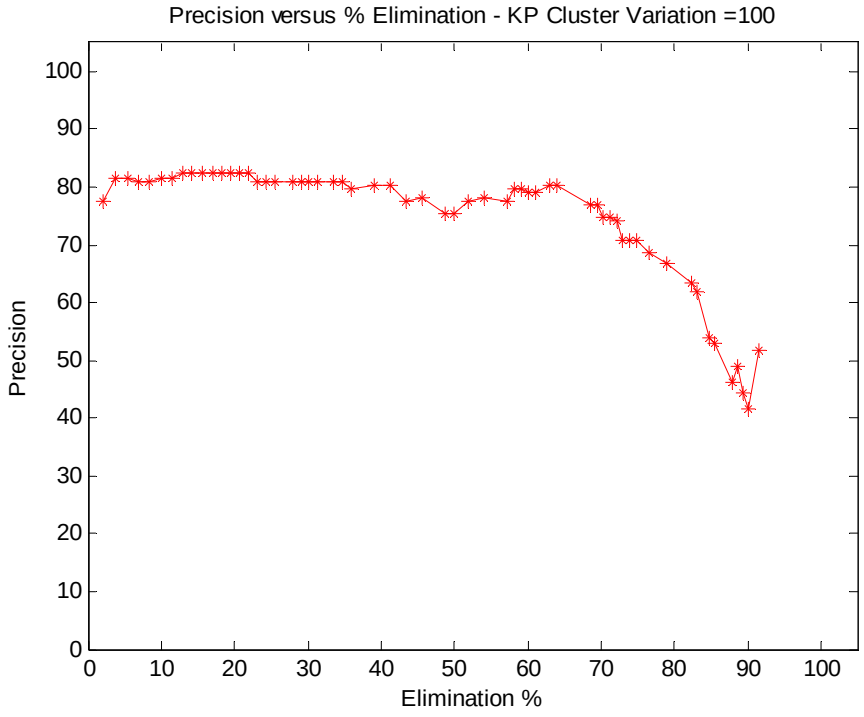
Figure 5.6. Performance of SIFT and after outlier detection for COLD database.

Different values for the parameter Ψ (number of *feature categories*) are investigated in the information-theoretic evaluation (review section 4.6.4). Values between 100 and 2000 are tested for the total 14848 keypoints of the training dataset, and the subsequent effect of eliminating keypoints of relatively high-entropy values is monitored. Figures 5.7-a, 5.8-a and 5.9-a show the calculated Precision versus the different percentages of high-entropy keypoints elimination; Ψ = 100, 500 and 1512 respectively. All the figures show that almost a constant Precision is maintained up to 60% elimination percentage. After 60% elimination percentage, Precision starts decreasing. The 500 clusters, shown in figure 5.8-a, are considered an ideal choice for the parameter since the performance curve undergoes a smooth variation.

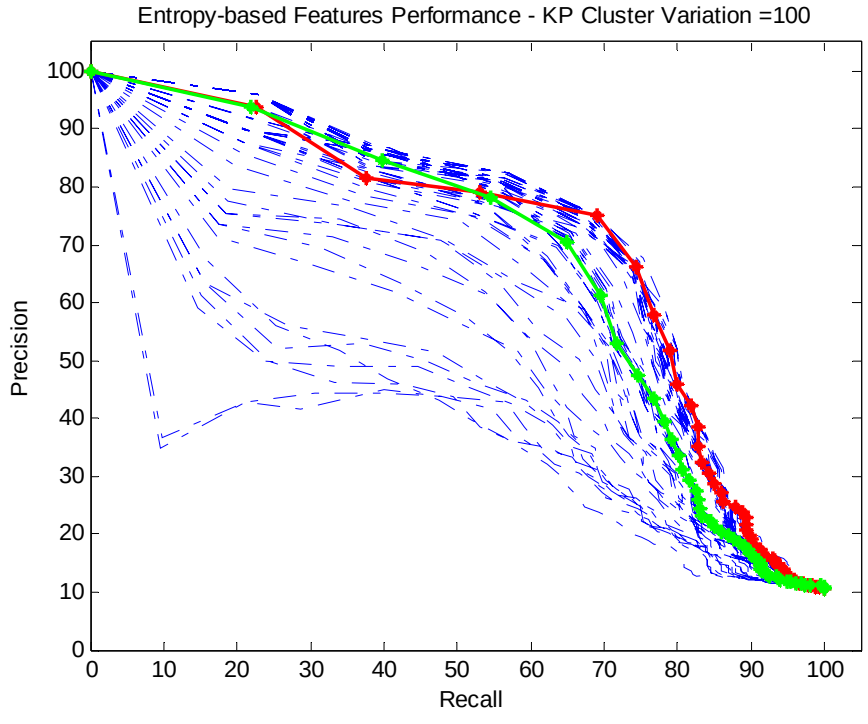
Subsequently, figures 5.7-b, 5.8-b and 5.9-b show the relation between Precision and Recall as a performance behavior for the whole retrieval process, and as a function of the different elimination percentages in figures 5.7-a, 5.8-a and 5.9-a respectively. In order to demonstrate the effect of feature evaluation, performances are also compared to original SIFT features after outlier detection (red curve). As the plots indicate, the relationships exhibit a dense bundle of curves, in which performance is close to and sometimes better than SIFT. Such bundle corresponds to the constant precision level indicated in figures 5.7-a, 5.8-a and 4.9-a. This means that localization accuracy is similar to the original SIFT. The green curve is the one selected for the codebook generation. It has a filtering threshold equal to 52% of high-entropy features, or reversely put, a 48% low-entropy feature set.

Figure 5.10 shows the localization performance of 48% low-entropy feature set (LEF) for Ψ = 100, 500, 1000, 1512, and 2000 for the second *test dataset*. The map contains on average 6886 keypoints. Low-entropy feature sets with cluster variations 100 and 500 show better performances than the original features extracted by SIFT and SIFT with outlier detection (OD). Higher cluster variations show less accuracy as data undergo extra division and lose meaningful content. It is worth mentioning that accuracy is higher in the test dataset compared to the training dataset, with a value of 93% for the filtered evaluated features (Ψ=100 and 500), versus 86% for original SIFT features.

Figure 5.11 shows the localization performance of five CB examples for the test dataset, which are generated from the previous entropy-based features examples. The map contains 575 keypoints, equivalent to 12 times compression than the entropy-based features map, and 28 times than the original SIFT map. Localization using the CB records much

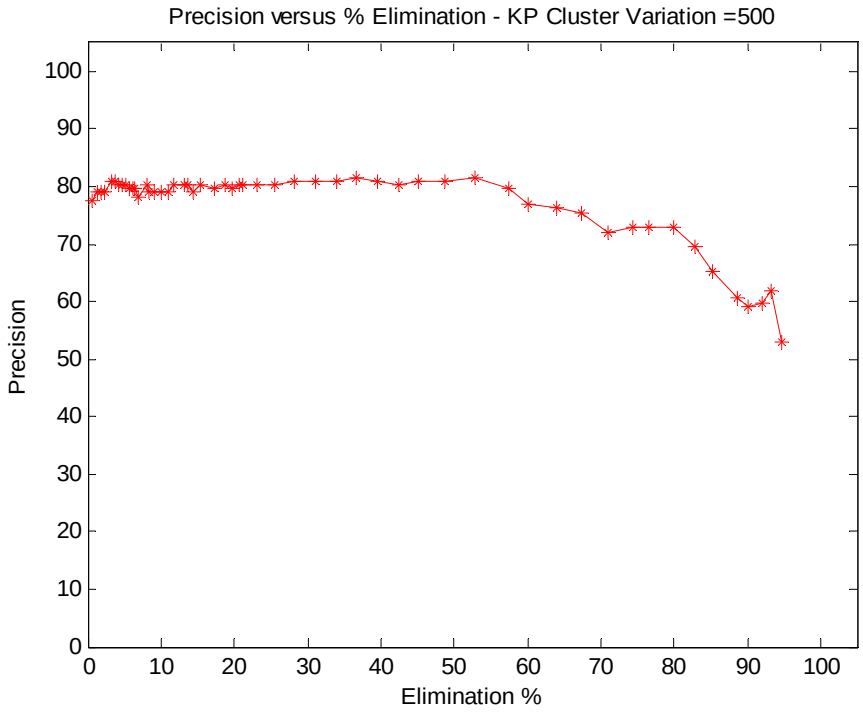


(a)

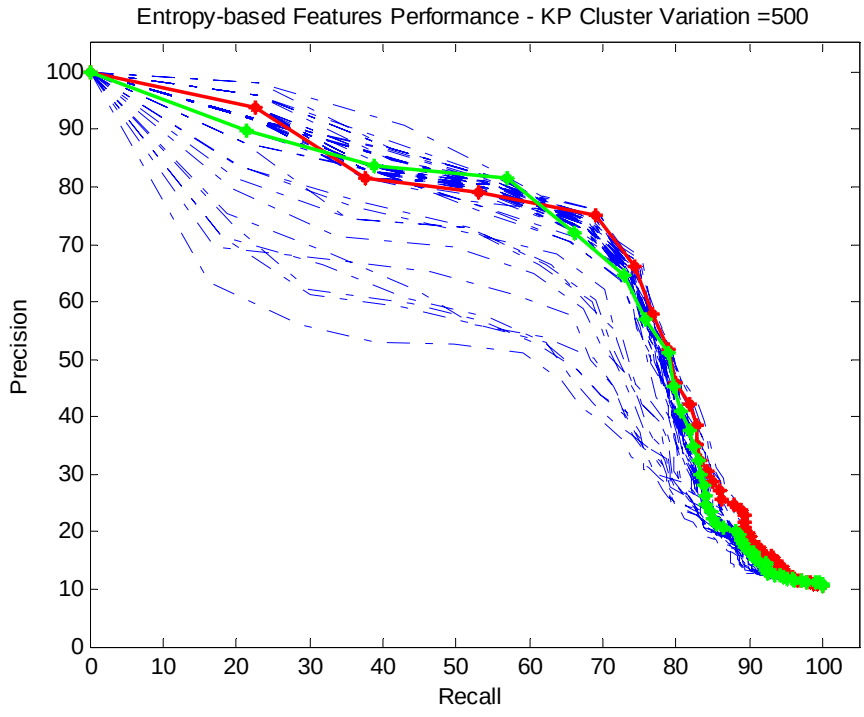


(b)

Figure 5.7. (a) Average localization Precision versus different percentages of high-entropy features elimination ($\Psi = 100$). (b) Average Precision-Recall performance for the different elimination percentages of figure (a). The red curve is the performance of original SIFT Algorithm (reference). The green curve shows an elimination percentage of 52% chosen for codebook generation.

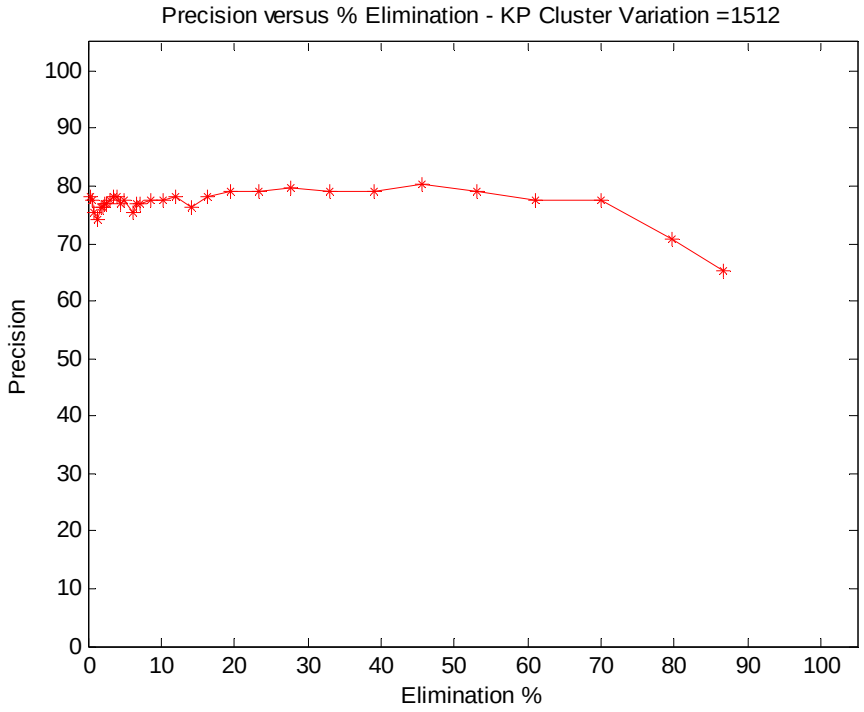


(a)

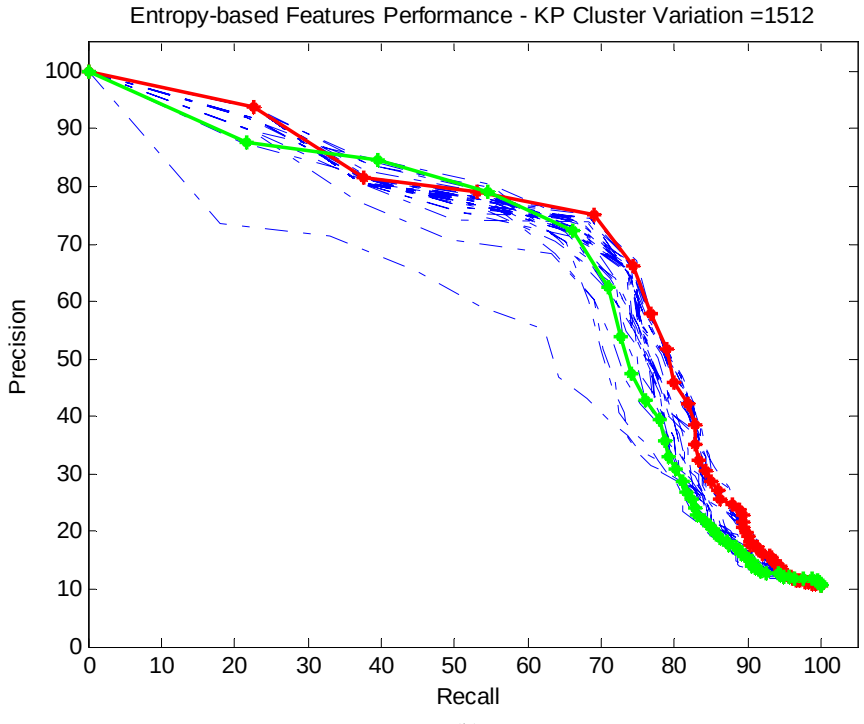


(b)

Figure 5.8. (a) Average localization Precision versus different percentages of high-entropy features elimination ($\Psi = 500$). (b) Average Precision-Recall performance for the different elimination percentages of figure (a). The red curve is the performance of original SIFT Algorithm (reference). The green curve shows an elimination percentage of 52% chosen for codebook generation.



(a)



(b)

Figure 5.9. (a) Average localization Precision versus different percentages of high-entropy features elimination (Ψ = 1512). (b) Average Precision-Recall performance for the different elimination percentages of figure (a). The red curve is the performance of original SIFT Algorithm (reference). The green curve shows an elimination percentage of 52% chosen for codebook generation.

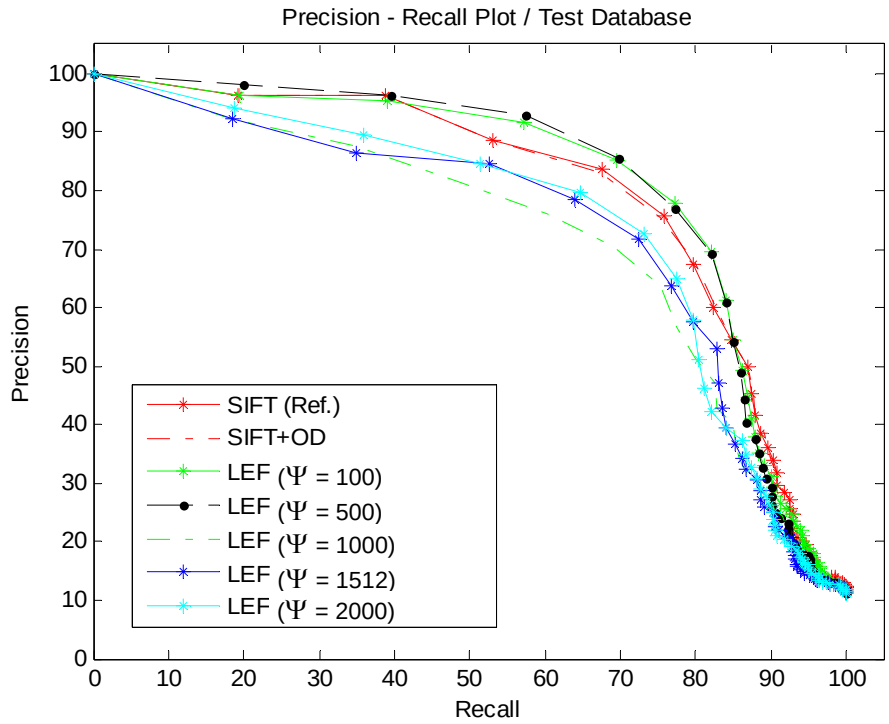


Figure 5.10. Average localization performance based on preserving 48% of low-entropy features.

lower performance than the entropy-based features, as well as than SIFT (57-64% accuracy). Hence, surprisingly, the CB fails to achieve the expected high accuracy of localization.

Table 5.3 summarizes localization performance for different employed maps. Measurements are obtained in Matlab/Windows environment on a P4, 3.2 GHz processor, and with the databases residing on external disk storage. As indicated, an average reduction in map size of 57% is obtained for the entropy-based feature map, and with better accuracy than the original SIFT map (90% for entropy-based features versus 86% for SIFT). The CB records 96% reduction in the map size, but at lower accuracy (62%). It records, however, acceptable accuracy at 20% Recall. Localization time is also reduced for the entropy-based map and CB by 33% and 54% respectively, compared to SIFT. This time includes 1.271 seconds, which is the SIFT execution time on the specified machine. Excluding this time, reduction in matching and retrieval times is linear with the storage reduction as proved in the previous chapter.

5.4 Discussion: Comparison with HEID and Approach Customization

The work presented in chapters 4 and 5 concerns a common proposed approach for generating efficient maps that contain the smallest set of relevant features for localization purpose. The

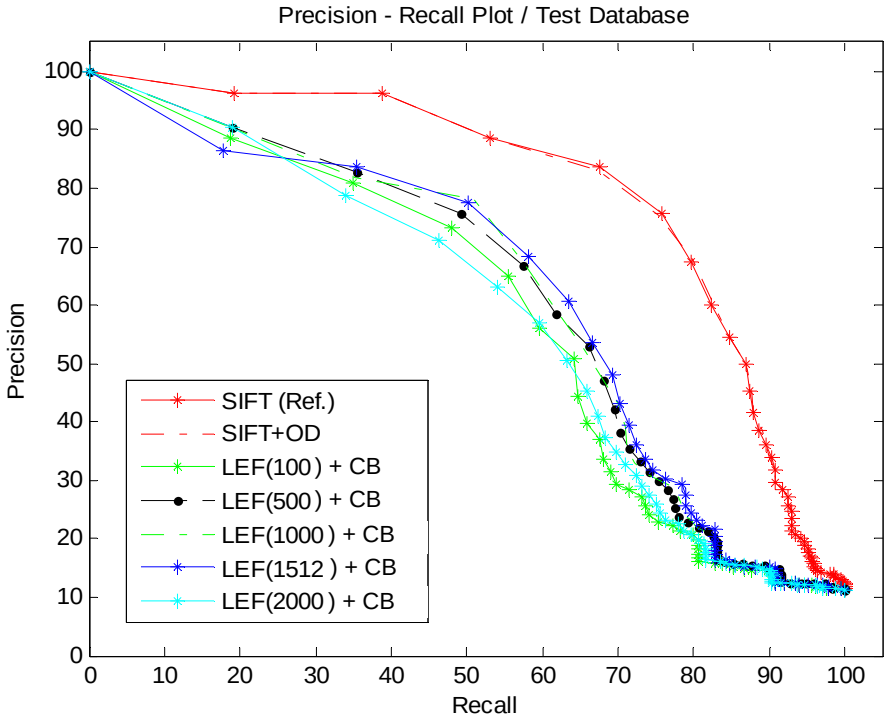


Figure 5.11. Average localization performance for codebooks generated from 48% of low-entropy features.

Table 5.3. COLD benchmarking database performance: Performance index versus feature map

Performance index \ Feature map	Original SIFT Alg.	Original SIFT + outliers detection	Entropy-based Features	Codebook
Map Size(Mbytes)	15.92	14.48	6.72	0.56
Average No. of KPs per image	333	302	140	12
Precision at 60% Recall (%)	86.19	85.93	90.84	61.95
Precision at 20% Recall (%)	96.15	96.15	97.09	89.86
Localization time (sec)	3.582	3.2271	2.2388	1.5440
Memory reduction (%)	-	9.07	57.77	96.47
Localization time reduction (%)	-	3.9	33.33	54.02

relevance of features is based on place discriminative categorization. An information-theoretic-based evaluation and selection of environment features is proposed. It attaches a discriminative feature set to a topological map in compressed format. The approach outputs two forms of features: filtered evaluated *entropy-based features* and filtered evaluated compressed *codewords*. An implementation for the approach has been presented in chapter 4

using HEID, the dataset acquired at the indoor environment of the Automation Laboratory at Heidelberg University. The robot setup carries a perspective camera of resolution 640x480. Wide-view images have been obtained by stitching sequential images together. This procedure generated panoramic images of average size of 1500x300 pixels. Testing the approach in the Heidelberg environment has shown outstanding performance using both feature forms. Localization performance using entropy-based features showed 96% accuracy with 68% reduction in space and 54% in time, while attracting performance was recorded using the codewords with 93% accuracy, 93% reduction in space and 78% reduction in time.

The COLD database investigations presented in this chapter did not show the same exact sound performance as with HEID. Initially, comparing the two datasets and the acquisition conditions in both, the following is summarized: The increase in the visual field of omnidirectional camera in COLD comes at the cost of image resolution when compared to the perspective camera used in HEID. A single image frame acquired by the limited-view perspective camera is exactly equal to the omnidirectional image frame acquired by the omni-directional camera. This accounts for a total lower resolution for COLD images. The resolution factor, in addition to large plain environment spaces (e.g. large wall areas), accounted for less details and pixel intensity variations in the images. The average number of features extracted per scene reveals this fact (333 for the COLD versus 1000 for HEID). A final difference between the datasets is that the COLD image set is of obvious poor quality.

Experimentations on the COLD database were successful as the Heidelberg database, but showed a different performance. It was found that entropy-based feature map outperforms the original extracted SIFT feature map. A localization accuracy of 90% has been recorded for the former, while managing to reduce more than 50% of the original map size and 30% of the localization time. The CB module provided high computational savings, but not at a high localization accuracy. It recorded 62% accuracy with 96% reduction percentage in the map size and 54% in the localization time for the distribution localization.

When comparing the results of the CB in the investigations carried out, it can be concluded that the codewords may not have represented the clusters of the entropy-based features in the most efficient way. The number of clusters is a sensitive factor for the quality of clustering. Less or extra division of data makes the resultant clusters miss or lose the

meaningful information content. From this perceptive, a thorough test is conduced using the Silhouette coefficient, $\overline{S}$ (equation 4.12), in order to identify the quality of clustering applied.

Figure 5.12 shows the values of the coefficient against the number of clusters. The relationship shows a monotonic increase that spans a small bounded range [0.02:0.17]. This indicates an accepted clustering quality over the studied range, and hence for the used clustering parameter. Consequently, it is deduced that other factors influence such performance more than the factor of the number of clusters, which puts forth customization limits according to the environment and sensor types.

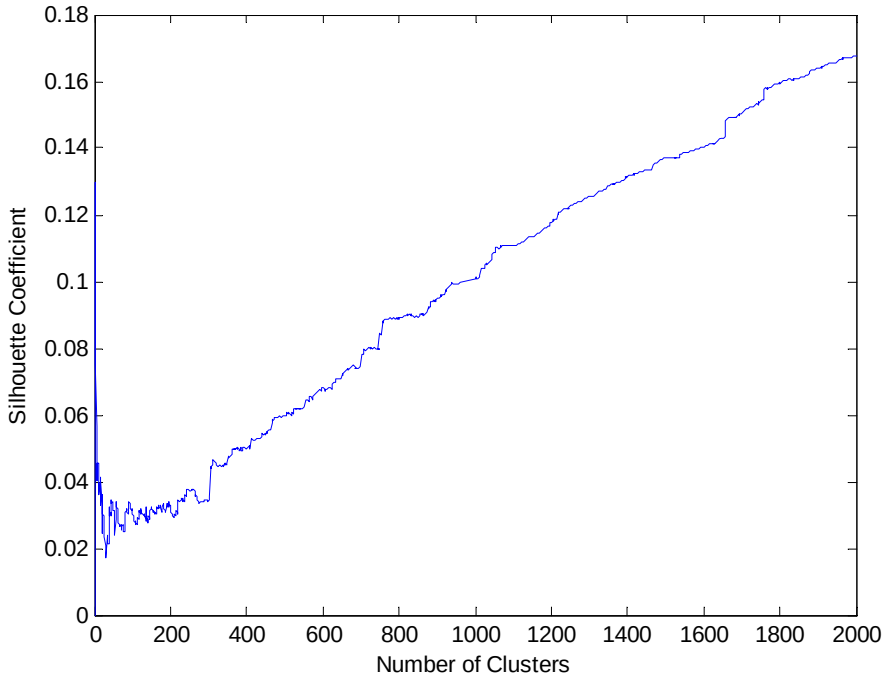


Figure 5.12. Average Silhouette coefficient against number of clusters.

The interpretation of the COLD database performance is as follows: It is clear that the images of the COLD datasets undergo conditions of severe illumination changes, poor quality, less resolution and contain less varying details. These conditions would possibly lead to less variation between the generated codewords. In the case that *only 12 codewords* are assigned per image to perform the scene recognition, the accuracy will naturally drop due to potential misrecognitions. The number of features and their descriptive content are practically insufficient in such case to do the recognition task efficiently. The details factor complies with

a previously mentioned constraint regarding the approach in section 4.4. The environment should contain moderate amount of details so as to achieve high localization accuracy. It is true that the approach tries to select data that maximize recognition and localization accuracies, but the environment might not assist achieving this goal if it contains data that are mostly redundant over the whole environment or data that have a small variation domain.

Within the experimentations, localization has been judged as an image retrieval system. The localization accuracy is set to be the precision at 60% Recall, which puts a restriction that more than half of the images bound to the best match should be retrieved. Such a setting can be used to derive a probability distribution over places, which allows fusion with tracking filters using multi-hypothesis. It is also still appropriate to set the accuracy to lower Recall values and obtain a single place solution (best match) in case the recognition rate is high. For example, referring to table 5.3, it is seen that Precision at 20% Recall provides high localization accuracy, and there is no need to track other retrievals (97% for entropy-based features and 90% for codewords). It is noted that the high percentage of Recall was imposed in order to validate the retrieval quality in general when employing features for recognition or to derive a location distribution. The comparison between the HEID and COLD datasets is summarized in table 5.4.

The same exemplar benchmarking database tested in this chapter has been assessed in [Ullah et al., 2008] using Harris-Laplace, SIFT and support vector machines. The authors report a recognition and robot topological localization rate of 88%. This is for a common adopted condition between the two works that same weather conditions are used for both training and testing. The recognition rate can be fairly compared to our attained performance at 20 % Recall, as the authors do not use distribution analysis as in our case (i.e. consecutive retrievals after the first best match). The comparison indicates suitability of the proposed approach for application.

In conclusion, a general customization for the approach can be applied. Either feature evaluation and compression components or the feature evaluation component alone can be employed. A CB performance is function of the size and variation of extracted features (i.e. environmental details). The approach proves general efficiency regarding accuracy and complexity and scales well to environment space as it has a topological context. The

localization performance is also tunable with respect to different performance criteria according to that required by the application.

Table 5.4. Comparing Heidelberg dataset to COLD dataset – Performance index versus feature map

	HEID dataset			**COLD dataset**		
No. of topological nodes	**7**			**9**		
Feature map / *Performance index*	**Original SIFT Alg.**	**Entropy-based features**	**Codebook**	**Original SIFT Alg.**	**Entropy-based features**	**Codebook**
No. of KPs per image	1000	320	67	333	140	12
Map Size(Mbytes)	40.07	12.81	2.67	15.92	6.72	0.56
Distribution localization - Precision at 60% Recall (%)	95.5	96.9	93.6	86.19	90.84	61.95
Best match localization - Precision at 20% Recall (%)	100	100	100	96.15	97.09	89.86
Localization time (sec)	22.81	10.36	4.96	3.58	2.24	1.54
Memory reduction (%)	-	68	93.3	-	57.77	96.47
Time reduction (%)	-	54.6	78.3	-	33.33	54.02

5.5 Summary

This chapter evaluated the information-theoretic modeling and localization approach presented in chapter four against a benchmarking database, specifically targeted for robot topological localization and map building testing. The results obtained have shown efficiency of the proposed approach and methods regarding the feature evaluation and compression components. This indicates the validity of the approach.

When high localization accuracy is an essential requirement, the modeled environment should necessarily possess an acceptable amount of details with acceptable variations. This factor additionally assists efficient codebook construction.

Since the proposed solution approach proves repeated proficiency in assisting the construction of information-rich maps and efficient topological localization, the developed topological model and localization module can be safely integrated into a hierarchical localization framework for metric navigation purposes. This will be introduced in the following chapter.

Chapter 6

Hierarchical Framework for Localization

This chapter presents a robot metric localization solution. It introduces an extension for the work presented in chapter 4, such that metric model building and metric localization are also supported. A hierarchical localization framework is proposed, together with a modified map that employs hybrid data. The hierarchy and hybrid data provide dual localization modalities (topological and metric) with high accuracy, scalability, as well as computational efficiency for the localization solution. The publications [Rady et al., 2010b; 2011] summarize this work.

6.1 Introduction

Hybrid localization provides topological and metric position estimation in a single framework. In those frameworks, the metric localization executes faster compared to traditional global approaches if it is processed hierarchically. This is because the searchable space is projected on certain topological region(s) and becomes smaller. Generally, the goal of hierarchical approaches and frameworks is two-fold: (1) Providing an organized data or information structure. (2) Reducing the set of candidates towards a solution, and hence, possible high computations are evaluated only on a minimal subset rather than the entire reference set.

Therefore, hierarchical processing systems are considered computationally efficient, and scale well with large data sizes. Despite such attractive performance, it is crucial to outline that the topological localization step preceding the metric one is critical in hierarchical localization frameworks. The topological step should be highly accurate, or otherwise, the robot will localize itself in the totally incorrect metric location, and will act based on that inaccurate estimation, which can be critically unsafe. A possible solution for such a situation (i.e. lower accuracy of topological localization) is to integrate filters which can track multiple hypotheses for the nodes. However, this does not controvert the fact that high topological localization accuracy or limiting the hypotheses will accelerate the localization.

In chapter four, an information-theoretic topological environment modeling approach has been introduced. The approach extracts a minimal set of relevant entropy-processed features and compresses them. It has shown high localization accuracy besides computational efficiency on the topological level. In this chapter, a localization framework is proposed, in which the topological implementation acts as a first level in a hierarchy, whereas a second level of hierarchy extends from the first, providing a 2-level hierarchical framework for hybrid localization. The accuracy performance of the second metric level is safe thanks to the accuracy provided by the prior topological level, because of the information-theoretic modeling approach applied. The top-down designed frame will provide dual localization modalities: coarse topological and fine metric. This is advantageous, since the robot does not have to work on the exhaustive metric level all the time it is executing its mission.

The specific integration of topological structure into hierarchical frameworks for localization purpose is summarized in the following pros. Firstly, the prior topological node identification will shrink the metric search space, either to a single space or a distribution over it, which is in all cases smaller than the total search space under investigation. Consequently, the global metric localization complexity is minimized. Secondly, attractive advantages of the topology will be introduced, such as scalability of the localization solution to large-spaces, which is a characteristic of topological models; introducing context and semantics into the entire system by assisting mapping functional places as well as modeling objects and memberships; solving loop closure by detection of previously visited places; and recovering from serious errors of lost or kidnapped robot situations. In the lost or kidnapped situations, the robot does not have to be physically teleported in another place. This scenario simulates

that the robot system might have undergone an internal error or reset operation that has issued the wrong position estimate. Thirdly, from the design point of view, the metric data will be locally assigned in each node, and not on a global level of the whole map. This provides significant robustness against data (feature) correspondence problem.

Several global methods can be implemented at the metric localization level. For example, the most widely applied MCL can be chosen. Nevertheless, we chose another alternative which is non-probabilistic triangulation techniques. The choice is not in regard to their favor, but for establishing a solution based on a unified homogeneous feature set. The same topological feature set that comprises non-geometric information will be augmented with additional position information to account for the metric features. The metric features will be used to resolve the robot's position through triangulation. In this view, the framework preserves a single hybrid feature-map and shares it between the two levels of the hierarchy. The information in the map is viewed as possessing two resolutions, where the coarse resolution is equivalent to the compressed codewords and is used to resolve the topological location, whereas the fine resolution is equivalent to the entropy-based features tagged with their additional positional information and are used to resolve the metric position. An additional reason in preferring triangulation techniques to MCL is related to the willingness to exclude dead-reckoning information (i.e. odometry) from the localization solution.

The structure of this chapter is as follows: In section 2, the modified problem formulation and solution structure meeting hybrid map building and hierarchical localization are given. The description of general triangulation and vision-based triangulation problems are presented in section 3. The triangulation solution using photogrammetric model and both iterative and geometric methods is given in section 4. For accurate and robust solution, detection and elimination of environment dynamics and mismatches, as well as feature selection are introduced in section 5. Section 6 is dedicated to the metric map construction, experimentation and localization evaluation and finally section 7 summarizes the work.

6.2 Modified Problem Formulation and Solution Structure

Since the chapter introduces an extension of the information-theoretic modeling approach of chapter four, the problem statement and the solution will be reformulated to adapt to the new hybrid solution. Stress is given to the *metric* problem statement.

6.2.1 General Problem Formulation and Solution Approach

Environment perceptual aliasing and correspondences are problems facing both topological and metric models, leading to a consecutive negative influence on correct localization. For this, characterizing the environment through discriminating features is a solution towards those problems and at the same time combines accuracy of feature and node identification. A second problem, concerning metric localization in particular, is the computational processing which scales up with the environment size. If the total space is processed in large operating environments, high computational processing is needed in identifying the pose. Such problem can be solved through a hierarchical processing structure for localization. In this structure, a topological localization is embedded in the first layer of the hierarchy and a second metric localization is conducted in the second layer of the hierarchy using the projected space by the first layer only. The metric localization execution will be faster by a factor equal to the number of topological places defined by the first level. The implementation of the first level, however, should be based on efficient node representation and identification in the suggested framework. That is to say the topological perceptual aliasing should be minimized.

Consequently, a 2-level hybrid framework founded on a hierarchy principle for attaining lower complexity is proposed. An efficient model building process is used in conjunction, in order to generate an information-rich map that solves the aliasing problems. The proposed solutions define the two following phases: hybrid feature-map construction and hierarchical localization.

A– Hybrid Map Construction

Given: (1) A set of environment significant places $N = \{N_1, N_2, ..., N_n\}$ of size n;
(2) A mobile robot equipped with a sensor capable of capturing a pattern of multiple features for every place $N_i, \in N$; $i=1,...,n,$, or a combined sensor configuration to fulfill this condition. The sensor should be also capable of measuring bearings or an additional sensor is employed for this purpose;
(3) Let N_i be described by a set of features S_i of potentially different size per place. The whole feature space is denoted by F so that $S_i \subset F$.

Required: (1) Construct an environment topological feature-map $\mathbf{M}^t$ (N, C, f^*) in the

form of undirected graph $T:=(N,C)$, where N forms the graph nodes, and C is a set of ordered pairs indicating the spatial interconnection between nodes N_i and N_j; $i,j=1,...,n;\ i \neq j$. $\mathbf{M}^t$ comprises a minimal feature set per place $f_i^* \subset S_i$, which corresponds to a minimal feature set in the map $f^* \subset F$; $f_1^* \cup f_2^* \ ... \cup f_i^* ... \cup f_n^* = f^*$, such that the general node identification probability $p(N_i \mid f_i^*)$ is maximized;

(2) Expand $\mathbf{M}^t$ into a hybrid topological-metric feature-map $\mathbf{M}^h$ which includes additional data that resolves the metric location. The map comprises geometric and non-geometric descriptors for the filtered set f^*: $\mathbf{M}^h(N,\ C\ ,\ f^{ng}\ ,\ f^g)$, with f^{ng} being an arbitrary non-geometrical descriptor selection and f^g being the geometric descriptor in the form of position vectors $\mathbf{p}^{\mathbf{im}}_{\mathbf{A}} = [X_A^m\ Y_A^m\ Z_A^m]^T$; $m=1,2...,M;$ for M features in node i.

Assumptions: (1) No a priori environment specification is given (objects, landmarks);

(2) No a priori knowledge about previous positions of the robot;

(3) The environment possesses a moderate or high amount of details;

(4) Scene dynamics and varying illumination environmental influence.

B– Hierarchical Localization

Given: (1) The hybrid feature-map $\mathbf{M}^h(N,\ C\ ,\ f^{ng},\ f^g)$;

(2) The extracted sensory pattern $f^{ng'}$ for an unknown topological node p^t;

(3) The angle measurement vector between the unknown robot heading direction and the visible features (i.e. bearings) $\boldsymbol{\alpha}$;

Required: (1) Estimate the robot's most probable topological node(s) p^t; $p^t \in N$, using the non-geometric descriptor part of the map;

$$p^t = g(f^{ng'}, f^{ng},\ N) \tag{6.1}$$

(2) Estimate the robot's metric position: $\mathbf{q_r} = [X_r\ Y_r\ \theta_r]^T$ in 2-dimensional space of the reference global frame of p^t, using the non-geometric and geometric descriptors, as well as the bearings to the k identified visible features:

$$\mathbf{q_r}(X_r, Y_r, \theta_r) = h\ (f^{ng'}, f^{g'}, \boldsymbol{\alpha}) = \bar{h}\ (\mathbf{p}^{\mathbf{i1}}_{\mathbf{A}}, \mathbf{p}^{\mathbf{i2}}_{\mathbf{A}}, \mathbf{...}\ \mathbf{p}^{\mathbf{ik}}_{\mathbf{A}}\ , \alpha_1, \alpha_2, ... \alpha_k) \tag{6.2}$$

where $f^{ng'}$ is required to resolve the identities of features in the estimated topological node and $f^{g'}$ are their geometrical descriptors.

The solution approach is based on similar structural components introduced in section 4.4, which maintain the same design properties and aspects. The new hybrid solution, however, includes the following modified components:

(a) **Hybrid feature extraction** for extracting topological features and metric clues that represents the features' metric positions *(i.e. hybrid feature-map).*

(b) **Hybrid localization** for providing either topological or extended hierarchical metric localization modalities according to the need.

The previously introduced components in the topological solution – optional preprocessing, information-theoretic evaluation and codebook – remain a main core for the general hybrid solution structure.

6.2.2 Hybrid Solution Structure

Figure 6.1 shows a modified structure for the solution presented in figure 4.5, such that it adapts to the insertion of additional metric model building and localization components. The hybrid feature-map generation undergoes the same concept provided in the topological model building and localization, except that a hybrid map is generated fusing both topological features and metric clues (figure 6.1-a).

Figure 6.1-b introduces the new hybrid structure. The first modification is in the feature extraction. It is modified into the default feature extraction module (SIFT in the proposed implementation), plus an additional feature localization module to estimate the metric location of features. The feature localization subcomponent can be implemented using disparity-based methods [Jebara et al., 1999; Min et al., 2007; Torr, 2002]. The methods employ stereo or perspective vision and implement structure from motion strategies to calculate depth information. Triangulation is applied to calculate the features' position relative to the robot, which can be later transferred to a global frame of reference.

The output features from the previous hybrid feature extraction is a feature set that possesses a non-geometric characterization together with its positional information in 3D.

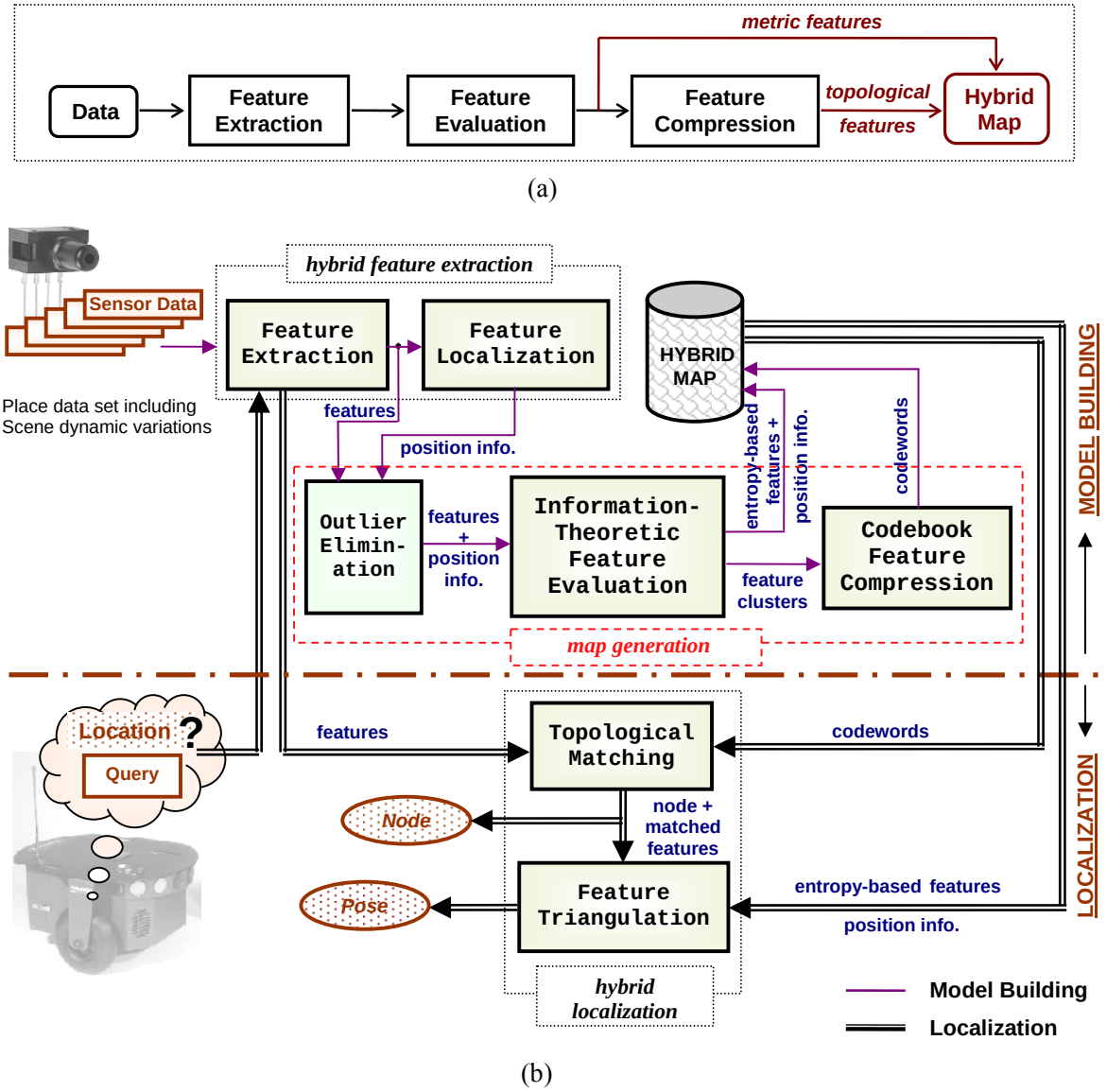


Figure 6.1. Modified solution structure for adapting into a hybrid model building and localization framework.

Features will follow the same map generation procedure of the previous design, including filtering and compression processes, before being stored in the final map. Entropy-based features with position information will account for metric data, while compressed codewords will account for topological data. The two formats together represent a single hybrid map, and in this sense, it contains location information in two resolutions.

The second modification in the structure is in localization. The previous matching component is replaced by a hybrid localization component, which includes the default

topological matching, besides an additional feature triangulation module. The latter module uses a photogrammetric projective model which relates the real-world to the image coordinates. Since triangulation is based on static positions, the accuracy of this module is ensured through dynamics and mismatches detection function. A second function is added for feature selection. While the first function overcomes the possible problems of data association and the environment dynamics, the second function will increase the precision of localization.

The topological localization executes the same as described previously. It is done via place matching using currently extracted features and the stored codewords. Accurate and fast local place (or distribution over places) recognition can be conducted. The metric localization always begins by executing topological place identification. After the place has been identified, the entropy-based features with their metric clues are utilized. The identity of features is resolved in a second matching, and the features' positions are triangulated to give the robot pose. The topological and metric scenarios can be traced on figure 6.1.

One remark about the feature localization is worth mentioning in the modified solution structure. In the figure, it is shown that the whole extracted data undergo this process of feature localization. This means that all features are involved in their localization processing, although a high percentage of them will be eliminated later on. This case, however, allows the robot to perform a single exploration round in the environment to generate the hybrid data. The second option is that the robot extracts the topological data (without the metric data) and goes through the evaluation and compression processes of the map generation. A second exploration round will be needed, in order to find the features' position data, in which the sensed data are compared to the processed entropy-based features, and then their position data are correspondingly fused. Finally, the metric data together with the topological one will form the final hybrid map. The second explained scenario has been adopted in our work.

6.3 The Triangulation Problem

Triangulation with active or passive landmarks is a robust and flexible absolute localization method. Different triangulation solutions exist [Cohen and Koss, 1992], in closed-form such as Geometric Triangulation and Geometric Circle Intersection, or as numerical iterative methods such as the Iterative Search and the Newton-Raphson method.

Closed-form solutions provide a unique solution, which is as accurate as the provided data. They are, however, subject to some geometrical constraints. Geometric triangulation is simply based upon the geometry of the landmarks and the relative angles between them as viewed by the robot. Circle Intersection solves two circle equations as a function of landmarks and robots positions. The two-circle intersection should yield the positions of the common landmark and robot.

Iterative solutions provide approximated solutions. The Iterative Search method triangulates two landmarks, which is sufficient to produce an accurate solution if the orientation is known or ignored. The method next iteratively searches through the possible space of orientations, given every pair of triangulated landmarks. The correct orientation is deduced to be the one yielding three robot positions that are closest together, from which a mean value is taken as the estimate. A second iterative solution is the Newton root solver methods. They make use of prior knowledge through iterative neighborhood function approximations and error calculations until a solution converges based on minimization of the errors. Those methods are adequate because of the non linearity of triangulation equation model.

In the scope of the metric triangulation, two methods will be studied: the closed-form Geometric Triangulation and a modified version of the iterative Newton-Raphson, which is the Gauss-Newton method.

6.3.1 Pose Estimation from Bearings

The principle of triangulation is based on simple geometrical constraints. The location of a point in space can be deduced through angles between the heading direction of the unknown position and lines of sight to at least three different reference points (commonly defined targets). The technique is mostly applied in the 2-dimentional horizontal plane projection of the space. See figure 6.2-a for a three-landmark triangulation configuration.

Considering the virtual environment defined in figure 6.2-a, it is required to estimate the robot pose variables (X_r, Y_r, θ_r) in 2-D space. The environment has four landmarks[1], from which only three are visible to the robot at a time, A_k; $k=1,\ldots,3$. It is assumed that each of the landmarks is individually identifiable, and that the robot has a priori knowledge of their

[1] Landmarks are any distinguishable objects in the environments with known locations. In this chapter features and landmarks will be used interchangeably, indicating the same thing.

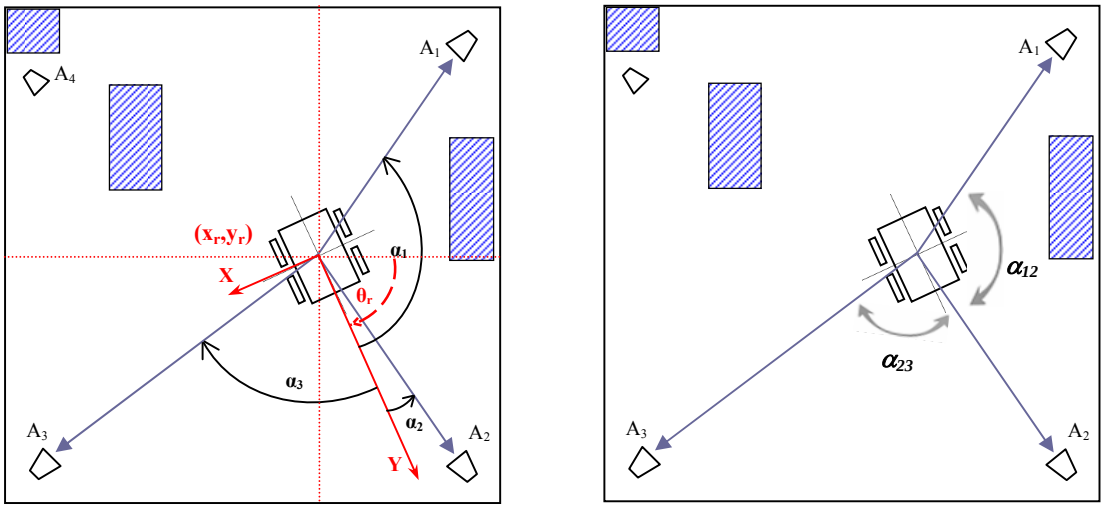


Figure 6.2. Three-landmark triangulation configuration. The position vector of the robot vector $p(x_r, y_r, \theta_r)$ can be determined by: (a) the angles defined from the robot heading to commonly defined targets or (b) the relative angles between the defined targets.

absolute positions in a global frame of reference. Each landmark can additionally have its own orientation. It is also assumed that the robot has a noisy sensor capable of measuring the direction to the landmarks. The output of the sensor is the direction and angle between the heading of the robot (robot's orientation) and the line joining the robot and the landmark. These angles are called the *bearings*, and are identified in the figure as α_1, α_2 and α_3 for landmarks A_1, A_2 and A_3 respectively.

For this scenario, the triangulation problem calculates the robot pose vector $\mathbf{q}(X_r, Y_r, \theta_r)$ as a non-linear function of the known positions of landmarks in the global reference frame, using either the measured angles (figure 6.1-a) or the angular separations (figure 6.1-b):

$$\mathbf{q}(X_r, Y_r, \theta_r) = \mathrm{g}(\mathbf{p}_{\mathrm{A}}^1, \mathbf{p}_{\mathrm{A}}^2, \mathbf{p}_{\mathrm{A}}^3, \alpha_1, \alpha_2, \alpha_3) \tag{6.3a}$$

$$\mathbf{q}(X_r, Y_r, \theta_r) = \mathrm{g}'(\mathbf{p}_{\mathrm{A}}^1, \mathbf{p}_{\mathrm{A}}^2, \mathbf{p}_{\mathrm{A}}^3, \alpha_{12}, \alpha_{23}) \tag{6.3b}$$

It can be assumed that each landmark position vector $\mathbf{p}_{\mathrm{A}}^{\mathrm{k}}$ includes the landmark's orientation θ_{lk} beside its position. This orientation can be set to zero in case the landmark appears identical when seen from all directions, such as with point features.

The landmark visibility factor is of primary importance in triangulation. In practice, it is difficult to have visibility of at least three physical targets at one time. Accordingly, using several point features as landmarks relaxes this constraint rather than using several object landmarks, since many of them are encountered together in the robot's view.

Triangulation using *bearings only* requires at least 3 identified landmarks to recover the full pose. Determining the robot's location and orientation from three landmarks is termed *three-object triangulation*. Some general geometric constraints limit the solution to a unique one. For example, when the robot lies on or very close to the circumference of the circle enclosing the three landmarks, or when the landmarks are collinear. The latter situation leads to two possible solutions, instead of a unique one. Another geometric constraint concerns the angular separations between the landmarks (see α_{12} and α_{23} in figure 6.2-b), which affects the accuracy of solution. The wider the bearings, the more accurate the triangulation. In other words, larger baselines connecting the landmarks are favorable. Hence, it is concluded that both the geometry and dispersion of landmarks as viewed by the robot are relevant factors that can suggest candidate landmark selection for the triangulation, since they control its accuracy.

6.3.2 Vision Bearings

Bearings can be obtained using range or vision sensing. As vision-based local features were proposed in the implementation of the initial solution structure, vision bearings are of concern. Figure 6.3 shows a simple perspective camera projection model. The image plane is located at a distance equal to the focal length f measured from the camera focal point. Objects in the 3D world are projected onto the image plane corresponding to the sensor chip.

Suppose (u_m, v_m) is the image location of a given landmark A in the real world. Assuming that the x-axis of the image plane, X′, is parallel to the robot plane of motion, then the bearings are a function of the pixel displacements from the principal point on the image plane and the camera focal length. A non-linear tangent function h preserves for example the horizontal bearing relation:

$$\alpha_m = h(u_m, u_0, f) \tag{6.4}$$

With the general pose definition in (6.3), the camera (and correspondingly the robot) position variables become a function of the landmarks positions in the object reference frame, the

projected image displacements and the camera focal length. The projected displacements – which are proportional to the angular measurements - are assumed to be disturbed by additive noise Δu. This noise can be modeled with a zero-mean Gaussian function. The presented camera model provides information on the angles to the landmarks only, with no information about the distance to the landmarks (i.e. depth information). Therefore, at least 3 landmarks are needed to identify the robot pose.

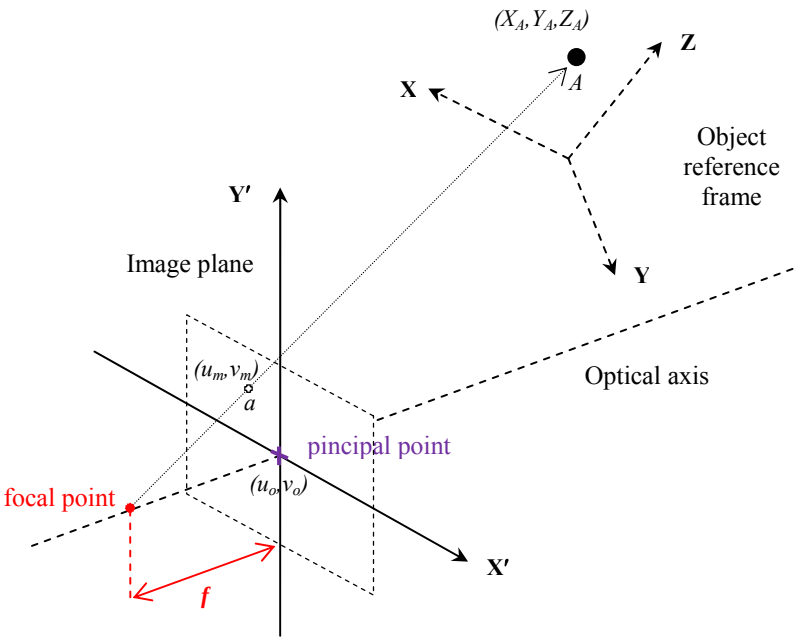


Figure 6.3. Camera model.

6.3.3 Triangulation Mathematical Formulation

The geometric localization in a general 3-D space notation is formulated as follows:

Given:

(1) A mobile robot located at a recognized node p^t in the environment topological graph T, with k visible and individually identifiable landmarks, such that $k \geq 3$ as shown in figure 6.2;

(2) The position vector for each landmark $\mathbf{p}_{\mathbf{A}}^{m} = (X_A^m, Y_A^m, Z_A^m)^{\mathrm{T}}$ in the metric frame of reference of p^t for $m=1,2\ldots,k$;

(3) The *bearings* $\alpha_1,\alpha_2,\alpha_3,\ldots\alpha_k$ as viewed from the robot, which relate the measured angles from unknown heading direction of the robot to the identified landmarks;

Assumption: The orientation of the landmark is ignored. This assumption implies that a landmark is identical when seen from any direction;

Constraint: Planar floor

Required: Estimate the robot geometric position in p^t: $\mathbf{q_r} = (X_r\ ,\ Y_r\ ,\ \theta_r)^{\mathrm{T}}$, in the 2-dimensional projective plane of the global frame of reference:

$$\mathbf{q_r}(X_r, Y_r, \theta_r) = g\,(\mathbf{p_A^1}, \mathbf{p_A^2}, \mathbf{p_A^3} \ldots \mathbf{p_A^k}\ , \alpha_1, \alpha_2, \alpha_3 \ldots \alpha_k) \tag{6.5}$$

where (X_r, Y_r) is the location in Cartesian space, theta is the robot's angular heading with respect to the world's positive *x*-axis, with a positive orientation taken in the counter clockwise direction, and *g* is a non-linear trigonometric equation of the given landmarks positions $\mathbf{p_A^1}, \mathbf{p_A^2}, \mathbf{p_A^3} \ldots \mathbf{p_A^k}$ and bearings $\alpha_1, \alpha_2, \alpha_3 \ldots \alpha_k$.

A three-landmark triangulation problem is a case of non-redundant information, where a single solution exists. For a number of landmarks larger than three, the case is over-determined, and an average numerical solution is obtained. Though landmark triangulation is not restricted to the vision-based detection of landmarks, we continue the detailed discussion of the solution in the next sections in a vision approach point of view.

6.4 Metric Localization using Triangulation

In this section, the solution-approach for the extended metric localization using triangulation is introduced. The approach is based on a single vision sensor and local point features implementation. Two methods for solving the triangulation problem are tested: the numerical iterative Gauss-Newton method and the closed-form Geometric Triangulation. The section begins by defining reference frames for the different physical components of the robot system. Next, the photogrammetric model is introduced, from which the equations describing the robot's pose are concluded. The model equations are solved with the iterative Gauss-Newton and Geometric Triangulation in subsections 4 and 5 respectively.

6.4.1 Geometry and Reference Frames

A body's position in 3-D space is defined by six variables: x,y,z describing the location of its center of mass, and what is called Euler angles (ω, ϕ, κ) describing the body's orientation.

The angles describe a sequence of rotations around the body's X, Y and Z axes, with respect to a fixed global frame of reference. The six variables form a vector called the *pose* of body.

For the robot pose estimation problem, different frame coordinate systems need to be defined according to the given physical components. Figure 6.4 identifies these frames for a situation of a moving robot with a camera sensor observing a stationary landmark. {W}, {V} and {C} are the reference coordinate systems of the global world, robot and camera respectively. It is often assumed that {C} has a fixed position with respect to {V} and aligns with it for non pan-tilt cameras. A constant offset vector $\mathbf{q_{off}} = (x_{off}, y_{off}, z_{off})^T$ shifts the camera a little away from {V}. The camera reference frame is chosen such that its origin coincides with the camera center (focal point) and with the Z-axis aligning with the camera optical axis.

In the figure, vectors **p** and **q** indicate the position of the tracked landmark and the position of the moving robot in {W} respectively, while **n** indicates the position vector of the landmark relative to the robot. The landmark position is always given in the camera frame {C}, which has in turn to be transformed to the robot and world frames through vector transformations. The metric localization problem solves for **q** = **p**-**n**, expressed in {W}. Locating the camera in {W} implies locating the robot in {W} as well. This is done through simple, and almost fixed, camera-robot transformation.

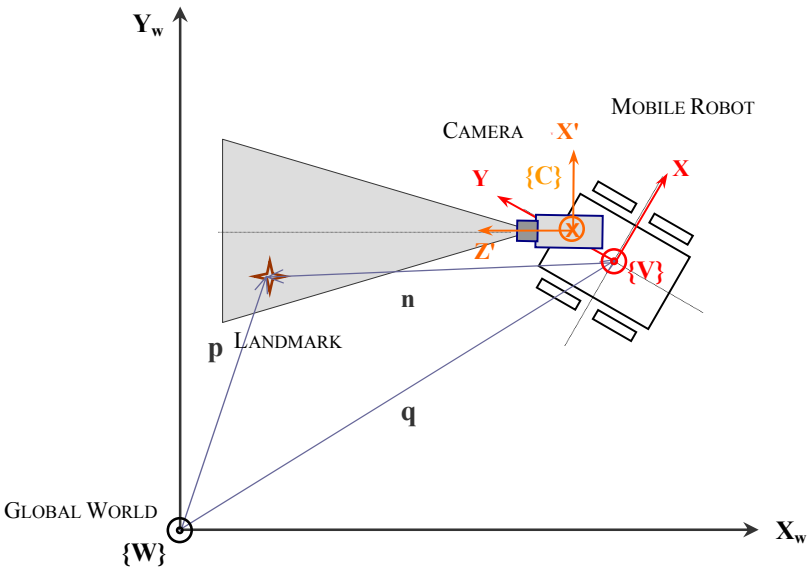


Figure 6.4. Reference frame coordinate systems-planar top view.

For a vector $\boldsymbol{\lambda}$ representing the orientation of {C} relative to {W}[2],

$$\boldsymbol{\lambda} = \begin{bmatrix} \omega \\ \phi \\ \kappa \end{bmatrix} = \begin{bmatrix} roll \\ pitch \\ yaw \end{bmatrix}, \tag{6.6}$$

a rotation matrix ${}^{\mathrm{W}}_{\mathrm{C}}\mathbf{R}$ is defined to be the transformation that maps vectors expressed in {C} relative to the global coordinate system {W}.

$${}^{\mathrm{W}}_{\mathrm{C}}\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{6.7}$$

The matrix can be defined in different order of rotations. For the rotation sequence *zyx*:

$${}^{\mathrm{W}}_{\mathrm{C}}\mathbf{R} = \mathbf{R}_{\mathrm{x}}(\omega).\mathbf{R}_{\mathrm{y}}(\phi).\mathbf{R}_{\mathrm{z}}(\kappa) \tag{6.8}$$

which is equal to:

$$\begin{bmatrix} \cos\phi\cos\kappa & -\cos\phi\sin\kappa & \sin\phi \\ \cos\omega\sin\kappa + \sin\omega\sin\phi\cos\kappa & \cos\omega\cos\kappa - \sin\omega\sin\phi\sin\kappa & -\sin\omega\cos\phi \\ \sin\omega\sin\kappa - \cos\omega\sin\phi\cos\kappa & \sin\omega\cos\kappa + \cos\omega\sin\phi\sin\kappa & \cos\omega\cos\phi \end{bmatrix} \tag{6.9}$$

From our current camera setup, the geometry can be simplified. The camera is fixed on the robot in a horizontal position (approximately), with its optical axis aligning with the robot's heading. This implies that the camera orientation variables ω and κ can be approximated to $\pi/2$ and 0 respectively. ϕ remains implication for the change of robot heading (i.e. the orientation θ_r). In the case that the landmark has a given orientation θ_{lm}, ϕ can be substituted with $\theta_r + \theta_{lm} - \pi/2$ instead.

6.4.2 The Photogrammetric Model

Photogrammetry is the technique of relating 3-D Cartesian coordinates of objects or simple points in real world to their corresponding 2-D image coordinates. The technique is mainly applied to derive 3-D position of objects from imagery. The fundamental model that has been used in the majority of photgrammetric applications is the *Collinearity equations* [Bossler et

[2] Orientation angles are expressed in Cardan notation which defines the angles as three combined rotations in different axes. Rotations are executed with respect to the resultant rotated axes and not to the initial axes.

al., 2002; Welch et al., 1991]. The Collinearity equations mathematically describe the fact that an object point, its corresponding image point, and the perspective camera center lie on a straight line under the idealized pinhole imaging model. The camera projection model in figure 6.5 shows this alignment, with the image plane set in the positive focal position (below the camera perspective center) instead of the real negative focal plane. The Collinearity equations can be applied differently to locate objects in the environment, locate the camera, or determine the camera intrinsic parameters. They summarize the mutual relations between image coordinates, object coordinates and camera parameters. In object-vehicle environments, their application captures the bearing measurements, in order to locate either the object (*i.e. map building, tracking*) or the vehicle (*i.e. localization*) through triangulation, by knowing the 3-D position of one to derive the other and with the knowledge of the camera intrinsic parameters. Single camera photogrammetry establishes the full 6 DOF relations (i.e. position and attitude) of an object with respect to camera, not only the 3 DOF, as long as the object being located has the suitable reference markings or control points in the camera view.

Figure 6.5 illustrates the projective mapping of a 3-D control point *A* to the 2-D image coordinates of the sensor plane. For a given vector of a control point relative to the camera $\overline{V_i} = \overrightarrow{cA}$, the corresponding projected space vector $\overline{v_i} = \overrightarrow{ca}$ is defined by the approximation:

$$\overline{\mathbf{v}}_m \approx \mathbf{P}\overline{\mathbf{V}}_m \tag{6.10}$$

where **P** is the Projection matrix describing the rigid camera transformation, and consists of a camera matrix **K** and a concatenation of a 3D rotation matrix **R** and a 3-dimensional translation vector **t**:

$$\mathbf{P} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}] \tag{6.11}$$

The projection matrix incorporates 6 *extrinsic* camera parameters in **R** and **t** for the complete description of the camera pose in 3D world (3 for 3D position and 3 Euler angles). The **K** matrix is called the Calibration matrix and has 5 free parameters. It contains the camera *intrinsic* parameters: the image format, the principal point and most importantly the focal length. A general form for the matrix is:

$$\mathbf{K} = \begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.12}$$

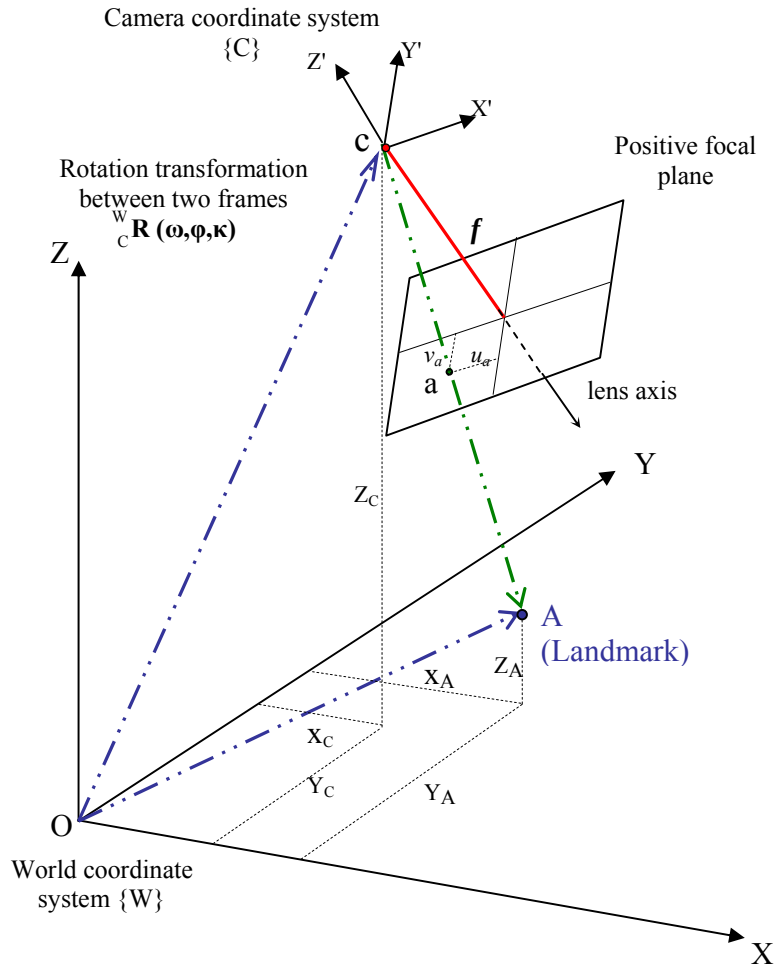


Figure 6.5. Photogrammetric projective model.

The parameters $\alpha_x = f / p_x$ and $\alpha_y = f / p_y$ represent the horizontal and vertical focal lengths in terms of pixels, where p_x and p_y are the pixel height and width respectively. γ is a coefficient accounting for the skew between the x and y axes of the pixel for non-rectangular pixels. It is often zero. u_0 and v_0 denote the principal point, which is the point where the optical axis intersects with the image plane, and which would be ideally at the image center. A distortion parameter can be also introduced to **K** (e.g. to account for camera radial distortions).

6.4.3 Derivation of the Robot Pose with the Collinearity Equations

As indicated in figure 6.5, X, Y and Z define the world frame axes, while X', Y', Z' define the camera frame axes, with Z' corresponding to the optical axis of camera. The camera is located

through its perspective camera center *C*, and the rotation matrix ${}^{\mathrm{W}}_{\mathrm{C}}\mathrm{R}(\omega,\varphi,\kappa)$ describing the camera's relative orientation in {W}, which are all required variables to estimate.

The 2-D image plane is located at *z=-f* from the camera frame, where f is the camera focal distance. In the photogrammetric model of figure 6.5, a landmark *A* in the 3-D world is projected onto the positive planar image surface corresponding to the sensor chip, resulting in an image point *a*. Projections in the X' direction are the proportional bearings measurements required to derive the planar position of camera and robot.

In the world coordinate frame {W}, a landmark positional vector $\mathbf{p_A}$ and the perspective camera center vector $\mathbf{p_C}$ are respectively defined by their position vectors as:

$$\mathbf{p_A} = \overrightarrow{OA} = (X_A, Y_A, Z_A)^T \tag{6.13}$$

$$\mathbf{p_C} = \overrightarrow{OC} = (X_C, Y_C, Z_C)^T \tag{6.14}$$

Within {C}, a projection vector of the landmark *a* is defined by:

$$\mathbf{p_a} = \overrightarrow{Ca} = (u_a, v_a, -f)^T \tag{6.15}$$

From vector relationships and coordinates transformation, it can be concluded that:

$$\mathbf{p_A} = \mathbf{p_C} + S.{}^{\mathbf{W}}_{\mathbf{C}}\mathbf{R}(\omega,\phi,\kappa).\mathbf{p_a} \tag{6.16}$$

where ${}^{\mathrm{W}}_{\mathrm{C}}\mathrm{R}$ is the camera-in-world rotation matrix between the two coordinate systems in the form of (6.7), and *S* is a scaling factor.

Rearranging (6.16), the landmark projection in the image frame becomes:

$$\begin{bmatrix} u_a \\ v_a \\ -f \end{bmatrix} = S^{-1}\,{}^{\mathbf{W}}_{\mathbf{C}}\mathbf{R}^{-1}(\omega,\phi,\kappa)\begin{bmatrix} X_A - X_C \\ Y_A - Y_C \\ Z_A - Z_C \end{bmatrix} = \frac{1}{S}\,{}^{\mathbf{W}}_{\mathbf{C}}\mathbf{R}^{\mathbf{T}}(\omega,\phi,\kappa)\begin{bmatrix} X_A - X_C \\ Y_A - Y_C \\ Z_A - Z_C \end{bmatrix} \tag{6.17}$$

Equation (6.17) is the specific form of the general equation (6.10), where the projection matrix **P** is equivalent to the scaling and rotation. The equation is regarded as a 3-D similarity transformation with seven transformation parameters including one scale, three orientation and three translation parameters. Nevertheless, the scale factor between the image and object vectors is different for each point, and hence analyzing the Collinearity equations

is not applicable for solving the scale in practical applications. Dividing rows 1 and 2 by row 3 in (6.17) eliminates the scale. The result is what is called the Collinearity equations [Bossler et al., 2002]:

$$u_a = -f\frac{r_{11}(X_A - X_C) + r_{21}(Y_A - Y_C) + r_{31}(Z_A - Z_C)}{r_{13}(X_A - X_C) + r_{23}(Y_A - Y_C) + r_{33}(Z_A - Z_C)} \tag{6.18a}$$

$$v_a = -f\frac{r_{12}(X_A - X_C) + r_{22}(Y_A - Y_C) + r_{32}(Z_A - Z_C)}{r_{13}(X_A - X_C) + r_{23}(Y_A - Y_C) + r_{33}(Z_A - Z_C)} \tag{6.18b}$$

which describes the angular measurements in both the horizontal and vertical directions in camera. It is noted that such an arrangement removes the nuisance scale parameter, but unfortunately transforms the equations into a more complicated and non-linear format.

Now, the camera and the robot poses can be derived from equations (6.18). Given at least three different landmark position vectors $\mathbf{p_A^m}$; *m=1,...,3,* (6.18) can be solved to derive the 3-dimensional vector $\mathbf{q_c} = (X_C, Y_C, \phi)^T$ of the planar pose of camera[3]. The robot pose vector $\mathbf{q_r} = (X_r, Y_r, \theta_r)^T$ is calculated through the camera pose and the camera offset vectors:

$$\mathbf{q_r} = \mathbf{q_c} - \mathbf{R_z}(\phi)\mathbf{q_{off}} \tag{6.19}$$

$$\begin{bmatrix} X_r \\ Y_r \\ \theta_r \end{bmatrix} = \begin{bmatrix} X_C \\ Y_C \\ \phi + \pi \end{bmatrix} - \begin{bmatrix} cos\phi & sin\phi & 0 \\ -sin\phi & cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{off} \\ y_{off} \\ 0 \end{bmatrix} \tag{6.20}$$

Determining the camera variables (sometimes referenced as camera exterior parameters), with the aid of known coordinates of world 3D control points and corresponding image coordinates is sometimes called *space resection problem*. The following section deals with the way to find a method for solving the complex Collinearity equations.

6.4.4 Solving Triangulation using Iterative Gauss-Newton Method

Solving for the robot pose requires solving the equations (6.18) of the photogrammetric model described in section 6.4.2. In these equations, the rotation matrix coefficients are

[3] The geometric pose estimation problem is discussed in the context of 2-D (planar) environment, which means that the location of the robot is given by its *pose* (i.e., position (*x, y*) and orientation φ). However, the concepts and results presented are equally valid for application in 3-D environments.

encapsulated as nonlinear elements, and this makes the problem inherently difficult to solve, and even more difficult to solve exactly. A solution, however, exists since the robot exists in the real world. An approximate solution can be assumed to be known, perhaps from the dead reckoning, which accelerates convergence towards a more accurate solution. For these two previous reasons, iterative methods can solve the problem. A traditional method to deal with the nonlinearity is linearization. There exist methods like Gauss-Newton and Levenberg-Marquart Algorithm (LMA) which can solve the function directly without linearization. Here, we apply the Gauss-Newton iterative method. Iterative methods have the disadvantage that they may converge to local minima when the given initial value is far away from the solution. However, if the initial value is nearby, a fast and correct convergence will occur.

The Gauss-Newton method is based on a linear approximation to the components of a given function $F(\mathrm{x})$ in the neighborhood of x. The method can only be used to solve non-linear least squares problems (NLLS). That is to say, finding $\mathrm{x} \in \Re^{\mathrm{n}}$ that minimizes an objective function in the form of a sum of squared function values:

$$F(\mathrm{x}) = \left\|\mathrm{r}(\mathrm{x})\right\|^2 = \frac{1}{2}\mathrm{r}(\mathrm{x})^T.\mathrm{r}(\mathrm{x}) = \frac{1}{2}\sum_{i=1}^{m} r_i(\mathrm{x})^2 \tag{6.21}$$

where $\mathbf{r}(\mathrm{x})$ is a vector of residuals; $\mathbf{r}$: $\Re^{\mathrm{n}} \rightarrow \Re^{\mathrm{m}}$ for given m functional relations $r_1, \ldots, r_m$ involving d variables $\mathrm{x}=(x_1, \ldots, x_d)$, with $m \geq d$.

The Gauss-Newton method applies a sequence of linear least squares approximations to the nonlinear function, each of which is solved by an "inner" direct or iterative process. In comparison to Newton's method and its variants, the algorithm is attractive because it does not require evaluating the second-order derivatives in the Hessian of the objective function.

Using the previously described photogrammetric model, it is required to estimate the robot position vector $\mathbf{q_r} = (X_r, Y_r, \theta_r)^T \in \Re^3$ from the appropriate angles to the landmarks at locations $\mathbf{p}_{\Lambda}^1, \mathbf{p}_{\Lambda}^2, \ldots, \mathbf{p}_{\Lambda}^m \in \Re^3$; $m > 3$.

$$\mathbf{q}_{\mathbf{r}}^* = \arg\min_{\mathbf{q}_r} \left\{\mathbf{R}(\mathbf{q}_{\mathbf{r}})\right\}, \tag{6.22}$$

Accordingly, the solution is formulated as NLLS as follows:

For m given equations, where m represents the number of observed landmarks, the general form of the non-linear equations (6.18-a,b) can be written as the two residual functions:

$$r_1^i(\mathbf{q_r}) = u_a^{\ i} + f\frac{U^i}{W^i} = 0;\ \ i = 1,2,...,m \tag{6.23}$$

$$r_2^i(\mathbf{q_r}) = v_a^{\ i} + f\frac{V^i}{W^i} = 0;\ \ i = 1,2,...,m \tag{6.24}$$

For any choice of $\mathbf{q_r}$, and with:

$$U^i = r_{11}(X_A^{\ i} - X_C) + r_{21}(Y_A^{\ i} - Y_C) + r_{31}(Z_A^{\ i} - Z_C) \tag{6.25}$$

$$V^i = r_{12}(X_A^{\ i} - X_C) + r_{22}(Y_A^{\ i} - Y_C) + r_{32}(Z_A^{\ i} - Z_C) \tag{6.26}$$

$$W^i = r_{13}(X_A^{\ i} - X_C) + r_{23}(Y_A^{\ i} - Y_C) + r_{33}(Z_A^{\ i} - Z_C) \tag{6.27}$$

Equations (6.23) and (6.24) can be written in the form of a 2 functional relation R^i to be equated to zero:

$$R^i(\mathbf{q_r}) = \begin{bmatrix} r_1^i \\ r_2^i \end{bmatrix} = 0;\ i = 1,2,...,m \tag{6.28}$$

In the neighborhood of $\mathbf{q_r}$, each functional R^i can be approximated with a Taylor expansion [Press, 1988]:

$$R^i(\mathbf{q_r} + \Delta\mathbf{q_r}) = R^i(\mathbf{q_r}) + \sum_{j=1}^{n}\frac{\partial R^i}{\partial q_r^j}\delta q_r^j + O(\Delta\mathbf{q_r}^2)\ \ ;i = 1,2,...,m \tag{6.29}$$

The matrix of partial derivatives appearing in the previous equation is the *Jacobian* matrix **J**, whose elements are:

$$J(q_r)_{ij} = \frac{\partial R^i}{\partial q_r^j} = \begin{bmatrix} \frac{\partial R^i(q_r)}{\partial X_r} & \frac{\partial R^i(q_r)}{\partial Y_r} & \frac{\partial R^i(q_r)}{\partial \theta_r} \end{bmatrix}; i = 1,2,...,m\ \&\ j = 1,...,3\,. \tag{6.30}$$

The Jacobian matrix of error functions with respect to parameters can be viewed as *m*x*3* matrix with scalar entries, or as an *m*x*1* matrix whose entries are vectors from $\Re^3$.

In matrix notation, the *2*x*m* residuals matrix $\mathbf{R} = (\mathbf{R_1}, \mathbf{R_2}, ..., \mathbf{R_m})^{\mathbf{T}}$ can be written as:

$$\mathbf{R}(\mathbf{q_r} + \Delta\mathbf{q_r}) = \mathbf{R}(\mathbf{q_r}) + \mathbf{J}.\Delta\mathbf{q_r} + O(\Delta\mathbf{q_r}^2) \tag{6.31}$$

By neglecting terms of order $\Delta\mathbf{q_r}^2$ and higher, the Gauss-Newton iterative equation is obtained as a set of linear equations for the corrections $\Delta\mathbf{q_r}$ that move each function closer to zero simultaneously.

$$\mathbf{R}(\mathbf{q_r} + \Delta\mathbf{q_r}) \approx \mathbf{R}(\mathbf{q_r}) + \mathbf{J}.\Delta\mathbf{q_r} \tag{6.32}$$

Setting the residual function, we want to find the updated vector parameter

$$\mathbf{q_r^{new}} = \arg\min_{\mathbf{q_r}} \left\{ \frac{1}{2} \left\| \mathbf{R}(\mathbf{q_r} + \Delta\mathbf{q_r}) \right\|^2 \right\} \tag{6.33}$$

From (6.31),

$$\frac{1}{2} \left\| \mathbf{R}(\mathbf{q_r} + \Delta\mathbf{q_r}) \right\|^2 = \frac{1}{2} \left\| \mathbf{R}(\mathbf{q_r}) \right\|^2 + \Delta\mathbf{q_r^T}\mathbf{J^T}\mathbf{R} + \frac{1}{2}\Delta\mathbf{q_r^T}\mathbf{J^T}\mathbf{J}\Delta\mathbf{q_r} \tag{6.34}$$

Differentiating (6.33) w.r.t. $\mathbf{q_r}$ and setting to zero,

$$\mathbf{J^T}\mathbf{R} + \mathbf{J^T}\mathbf{J}\Delta\mathbf{q_r} = 0 \tag{6.35}$$

The increment $\Delta\mathbf{q_r}$, which indicates the descent search direction for **R**, becomes the solution to the *normal equations*:

$$(\mathbf{J^T}\mathbf{J})\Delta\mathbf{q_r} = -\mathbf{J^T}\mathbf{R}(\mathbf{q_r}) \tag{6.36}$$

The assumption $m \geq 3$ is necessary, otherwise the matrix $\mathbf{J^T}\mathbf{J}$ is not invertible and the normal equations cannot be solved (at least uniquely).

The change is next added to the solution vector

$$\mathbf{q_r^{new}} = \mathbf{q_r^{old}} + \Delta\mathbf{q_r} \tag{6.37}$$

And the process is iterated to convergence.

With $\Delta\mathbf{q_r} = \mathbf{q_r^{new}} - \mathbf{q_r^{old}}$, equation (6.35) can be obtained equivalently by minimizing the sum of squares of the RHS of equation (6.31), this means setting it to zero, which would lead to:

$$\mathbf{J}.\Delta\mathbf{q_r} = -\mathbf{R} \tag{6.38}$$

The Gauss-Newton method can minimize a least square function in a few iterations. However, it does more work per iteration for the multiplications needed to form $\mathbf{J^T}\mathbf{J}$ and factorize it. In an exceptional case when $\mathbf{J^T}\mathbf{J}$ is singular, the Gauss-Newton method will fail. However, some $\lambda > 0$ can be chosen to obtain downhill search direction $\mathbf{q_r}$ from the *Levenberg-Marquardt* equations:

$$(\mathbf{J^T}\mathbf{J} + \lambda\mathbf{I})\Delta\mathbf{q_r} = -\mathbf{J^T}\mathbf{R} \tag{6.39}$$

The Newton method iteratively proceeds by first calculating **R** and **J**, with the initial guesses for X_r, Y_r, and θ_r of $\mathbf{q_r}$. For this set of equations, the Jacobian matrix can be approximated through finite differences.

$$J_{ij} = \frac{r_i(\mathbf{q_r} + \Delta\mathbf{q_r}) - r_i(\mathbf{q_r})}{\Delta\mathbf{q_r}^j} \tag{6.40}$$

Once **R** and **J** are calculated, the absolute sum of the **R** solutions is checked to be within some tolerance value. If so, then the initial guess has been accurate and the iteration terminates. If not, LU decomposition is performed to obtain new values of **R**. Again root convergence is checked, and if the sum of the absolute values of these solutions is within tolerance, the iteration terminates. Otherwise, the entire process is repeated until a possible solution is found or no solution is reported.

Solving equations (6.22) and (6.28) is regarded as an optimization problem that seeks minimizing the measurement errors defined by (6.23) and (6.24). The given set of observed landmark projections $\hat{\mathbf{v}}^i = (\hat{u}_a^{\,i}, \hat{v}_a^{\,i}, -f)^T$ can be modeled as theoretical image points $\mathbf{v}^i = (u_a^{\,i}, v_a^{\,i}, -f)^T$ perturbed by a measurement Gaussian noise e^j.

$$\hat{\mathbf{v}}^i = \mathbf{v}^i + \mathbf{e}^i \tag{6.41}$$

And consequently, the objective function to be minimized is:

$$\sum_{i=1}^{m} r_i(x)^2 = \sum_{i=1}^{m} \left(\hat{\mathbf{v}}^i - \mathbf{v}^i\right)^2 \tag{6.42}$$

which is equal to the summation of both residuals of R^i in (6.28),

$$\sum_{i=1}^{m} \left(\hat{u}_a^{\,i} + f\frac{U^i}{W^i}\right)^2 + \left(\hat{v}_a^{\,i} - f\frac{V^i}{W^i}\right)^2 \tag{6.43}$$

with U^i, V^i and W^i defined by equations (6.24-26).

Equations (6.42) and (6.43) conform with a general form for least-squares score function:

$$\mathrm{s} = \sum_{i=1}^{m} \left\| \hat{\mathbf{v}}_i - \mathbf{P}\mathbf{V}_i \right\|^2 \tag{6.44}$$

and is analogous to the projection defined in (6.10). The score function aims at minimizing the sum of the squared distances between the image coordinate $\hat{\mathbf{v}}_i$ and the 3D feature vector

$\mathbf{V}_i$. The cost function defines the sum as a squared error between the image values (the actual data) and the projected 3D model values (predicted values).

It should be noted that this solution is not restricted to estimating the position vector of the robot (feature triangulation module in the hybrid localization component). The same solution is applied in the model building phase for estimating the position vectors of features (feature localization module in the hybrid feature extraction component) for given calculated robot poses – refer to figure 6.1-b.

6.4.5 Closed Form Geometric Triangulation Solution

Triangulation can be solved using closed form solutions such as the Geometric Triangulation. In Geometric Triangulation, the computed pose is accurate as the data provided. The geometry for the solution is, however, not straight forward as multiple solutions of valid robot poses are allowed.

The computation of solution in Geometric Triangulation is not mathematically straight applied. The angles should be ordered properly to ensure that a unique solution is computed as stated in [Cohen and Koss, 1992]. The proper order constraint can be solved by imposing other easier constraints as done in [Esteves et al., 2003] to provide some flexibility. Geometric Triangulation works consistently with properly ordered landmarks, and with the robot located inside the triangle formed by the landmarks. There are areas outside the landmark triangle where the geometric approach works, but these areas are difficult to determine and are highly dependent on how the angles are defined.

Geometric Triangulation is solved by defining a set of angles, which are illustrated in figure 6.6, as follows:

$$\lambda_{31} = 2\pi + (\lambda_1 - \lambda_3) \tag{6.45}$$

$$\lambda_{12} = \lambda_2 - \lambda_1 \tag{6.46}$$

Define ϕ to be the angle between the positive x-axis and the line formed by the points of landmarks 1 and 2.

Define σ to be the angle between the positive x-axis and landmarks 1 and 3, plus ϕ.

Define the constants:

$$p = \frac{L_{31}\sin\lambda_{12}}{L_{12}\sin\lambda_{31}} \tag{6.47}$$

$$\tau = \tan^{-1}\left[\frac{\sin\lambda_{12} - p.\sin\gamma}{p.\cos\gamma - \cos\lambda_{12}}\right] \tag{6.48}$$

The distance from landmark 1 to the robot is calculated:

$$L_1 = \frac{L_{12}\sin(\tau + \lambda_{12})}{\sin\lambda_{12}} \tag{6.49}$$

Accordingly:

$$x_R = x_1 - L_1\cos(\phi + \tau) \tag{6.50}$$

$$y_R = y_1 - L_1\sin(\phi + \tau) \tag{6.51}$$

$$\theta_R = \phi + \tau - \lambda_1 \tag{6.52}$$

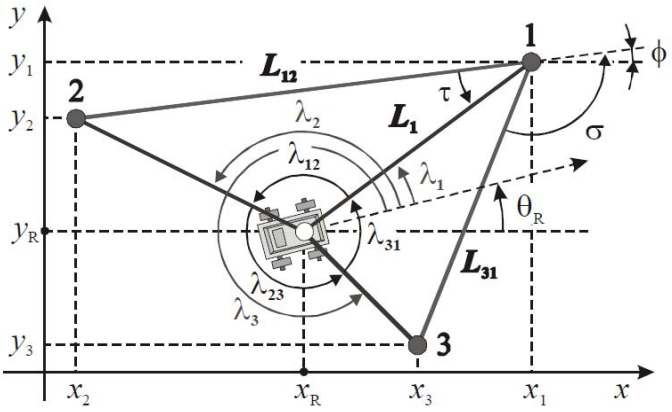


Figure 6.6. Defined angles for calculating the Geometric triangulation [Esteves et al., 2003].

6.5 Accurate Triangulation

The computational and accuracy performances of metric robot localization have been enhanced through the hierarchical processing and the improved topological localization respectively. A second requirement at this level is still to ensure accuracy and robustness of the solution. The main sources of metric localization errors are the following: (i) Mismatched features, (ii) Dynamic features, (iii) Features spatial arrangement, and (iv) Propagated errors from locations of features. Some solutions are suggested for the three errors sources.

Triangulation techniques assume that features are static, or else localization will exhibit errors. The given assumption of moderate dynamics influencing the operating environment will disturb the static state condition and will hinder realizing accurate and robust solution. Possible mismatched features have a similar effect on the triangulation solution as the environment dynamics. Therefore, a special functionality is introduced at the metric level for the detection of environment dynamic features and mismatches. A second functionality for proper selection of features through their spatial arrangement is introduced. This should minimize the imprecision in the estimated solution. The two introduced functionalities increase the robot localization accuracy. To implement them, two criteria are presented; one for the common detection and exclusion of dynamic features and mismatches, while the other for the selection of features based on geometrical layout.

6.5.1 Data Association and Environment Dynamics Detection

This functionality detects possible mismatched feature pairs that occurred at the topological level, when the current robot view is compared to the topological features of the map. Dynamic features may also exist if some environment objects are relocated to new places instead of their previously registered ones. Failing to specify the identity of feature because of multiple-occurrence or similarity metrics is the cause of mismatches. It is a common problem in recognition, known by data association or correspondence problem. In another view, the identity of a dynamic object can be correctly identified; however features of such an object carry false information for robot localization, and hence are not desired. On the one hand, those features should be eliminated from the map if it is required to maintain static features only for metric localization purpose. On the other hand, the features can be preserved for topological induction at the first level only, but they should be excluded at the second metric level. In any case, dynamic and mismatches situations should be detected and excluded before the triangulation is executed.

A mismatched or dynamic feature can be detected using spatial relationships between features. The following distance measure is utilized in order to classify those undesired features as outliers, and consequently, exclude them from the feature set recognized by the topological level. The distance measure has the form:

$$d_m = \left(\sqrt{(X_m - X'_m)^2 + (Y_m - Y'_m)^2}\right) \tag{6.53}$$

where (X_m, Y_m) and (X'_m, Y'_m) are the location of the m^{th} matched pair in the current image view and the image of the recognized topological node respectively; $m=1, 2, ..., M;$ and M is the number of matched features between the two images. With a simple splitting method like clustering or histogram generation for the calculated distances $D = \{d_1, d_2, ..., d_m\}$, outliers are easily encapsulated, and hence excluded. A small size for the number of clusters or the histogram bin size (e.g. 3-4) is sensitive enough to detect the outliers.

6.5.2 Feature Selection

In section 6.3.1, some factors have been mentioned as affecting the accuracy of triangulation (i.e. angular separation and collinearity of landmarks). Therefore, the features detected in the robot's camera will be filtered based on these two measures: the geometry of features in real world and the features' dispersion in camera view. The geometry measure will try to avoid quasi-collinear features, since localization error becomes large. The geometry measure is fused with another for the selection for widely dispersed features in the camera view. The dispersion measure contributes to less measurement errors in the bearings. Therefore, feature selection for triangulation will be based on the following weighted criterion:

$$d_q = \mu.Dispersion + (1-\mu).Non-collinearity \tag{6.54}$$

which combines the dispersion of features in the camera view with the non-collinearity of features. μ is a weighting factor.

Angular dispersion between three given features, j,k and m, will be defined as:

$$Dispersion(j,k,m) = \begin{cases} \dfrac{l_1}{l_2} * \dfrac{(l_1+l_2)}{w} & if \quad (l_1 < l_2) \\ \dfrac{l_2}{l_1} * \dfrac{(l_1+l_2)}{w} & otherwise \end{cases} \tag{6.55}$$

where l_1 and l_2 denote the horizontal displacements between the features, and w denotes the width of image view, both measured in pixels. In a similar manner the displacements l_1 and l_2 can be defined as vertical or Euclidian displacements. We have chosen to define horizontal displacements only in the implementation.

Non-collinearity will be defined as a function of two parameters, the perimeter of the triangle enclosing the three features and the aspect ratio of the three feature construction:

$$Non-collinearity(j,k,m) = perimeter(j,k,m) + aspect\_ratio(j,k,m) \tag{6.56}$$

where the aspect ratio is taken to be the reciprocal ratio between the maximum base of the enclosing triangle b to the corresponding height h, such that:

$$aspect\_ratio(j,k,m) = \begin{cases} \dfrac{h}{b} & if \quad (h < b) \\ \dfrac{b}{h} & otherwise \end{cases} \tag{6.57}$$

Figure 6.7 illustrates the parameters used in the measures of dispersion and non-collinearity.

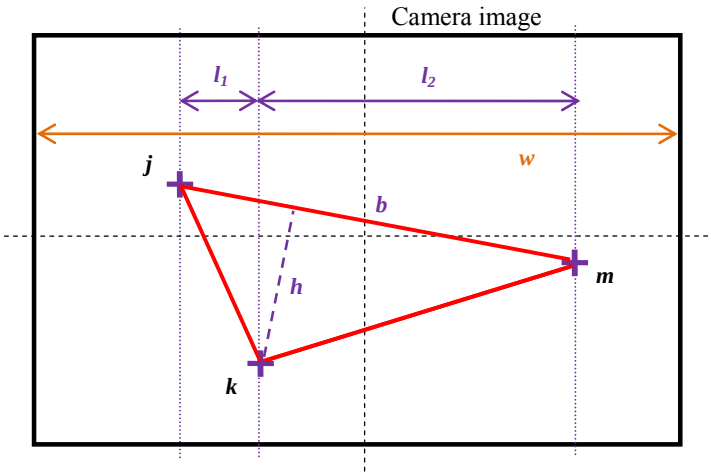


Figure 6.7. Parameters used in the metric feature selection criterion.

6.6 Experimentation and Results

The hierarchical localization evaluation is given in this section, with quantization for metric localization errors. A reference system is introduced for generating the ground truth of pose.

6.6.1 Ground Truth Reference System

In order to determine the geometric accuracy of the proposed hierarchical localization and quantify localization errors, the real pose of the robot needs to be known. For this purpose, the Krypton K600 system, a product of the company Metris (URL: [met]), is used as the

reference system (see figure 6.8). The system measures the position of objects in 6 DOF with high precision. It consists of a 3-infrared-camera system with a controller to calculate the 3D-position of infrared LEDs (so-called markers) by triangulation. These markers are attached to the movable body for tracking. The reference system offers an accuracy of up to 2 micrometer and can cover an area of ca. 17 cubic meters (refer to Table 6.1 for detailed technical information). To obtain measurements of fixed points as well, a Spaceprobe can be used. It allows determining the position of any point through direct contact.

Figure 6.8. Krypton system K600 used for generating ground truth values [URL_MET].

Table 6.1. K600 system technical specification

Space coverage	*17 m^3*
Resolution	*2 μm*
Accuracy	*Up to 60 μm*
Temperature range	*15-35 °C*
Relative Humidity	*0-80 %*
Acquisition frequency (using one marker)	*650 Hz*
Acquisition frequency (using three	*325 Hz*

The evaluation of the metric localization could not be applied successfully to the Heidelberg University environment using the reference system. The reference system was heavy to transfer, and exhibited difficulties in tracking the robot when positioned on its tripod, because the markers failed to be efficiently detected most of the time. For efficient

markers detection, the system camera should be hanged on the ceiling of the testing environment. Hence, the mounted markers are always in the camera view for every possible move executed by the robot (especially rotations). Therefore, the evaluation of the Heidelberg University environment has been carried out manually by recording the real geometric position by hand. A second evaluation using the reference system has been carried out in a laboratory, where a suitable hanging mechanism is afforded. This has enabled conducting several and thorough testing experiments to precisely quantify the metric errors, although this issue needed to create a new map.

6.6.2 Localization Evaluation for HEID

In this evaluation, the first level of hierarchy uses the codebook 'E' of table 4.5 as a topological map. It is based on preserving 36% of low-entropy feature set. The topological map size is 2.67 megabytes with an average of 67 features per place. The metric data size is 12.81 megabytes, with an average of 320 features per place. Both constitute the data to be processed in the hierarchical localization. A place matching is first executed at the first topological level, and triangulation is next executed at the second metric level using the detected features in the topological solution provided by the first layer.

Before triangulation execution, metric features are investigated using criterion (6.53), in order to detect dynamic features and mismatches. Figure 6.9 shows a test example in an office room of the Heidelberg University environment. The two images represent the current camera view as seen by the mobile robot and the best matched place image retrieved by the topological matching module. The paired detected matches by the matching module are shown in the figure. Using a histogram of three bins to classify the distances in (6.53), three outliers are detected, from which two are mismatched features, while the third is a feature for a relocated object. The test example has been recognized with a topological accuracy of 100% at 60% Recall at the default camera resolution (640x480 pixels).

To quantify the metric errors, the robot is guided through the environment to test localization in several places. At each place, the robot executes a rotation control through a joystick, stitches the images, and then runs the hybrid localization to determine its pose. In the iterative Gauss-Newton triangulation, the required initialization is assigned to the origin of the local frame of each topological node, with zero orientation (i.e. $(X_r, Y_r, \theta_r) = (0,0,0)$).

Figure 6.9. Dynamics and mismatches detection. From feature matching, features are classified into inliers that undergo the triangulation and outliers that are excluded. Outliers are identified in magenta and inliers in cyan. The upper image is the current camera view, while the lower image is the identified topological node.

The average localization errors recorded for Heidelberg office experiments are summarized in table 6.2. The average error in x and y position variables, as well as in the orientation, is recorded for the Iterative Gauss-Newton and Geometric triangulation methods. The Gauss-Newton method has been implemented to support more than three landmarks. On average, six robust landmarks are often detected and triangulated per single estimation. Due to this data redundancy, the results in the table record lower localization errors for the Gauss-Newton method in comparison to the Geometric Triangulation method.

Table 6.2 also compares the Geometric Triangulation using random features versus employing the feature selection criterion defined by equation (6.54). In this experimentation, the weighting factor is set to a value of 0.5. Results indicate that online feature selection based on geometry decreases possible localization errors. We note that both the robot and features ground truth were recorded by hand, which makes the process subject to an error range of a few centimeters. The table also records the metric localization time, which is bounded to a few microseconds execution time of the triangulation. The total hierarchical localization time is within 5 seconds for the given robot setup. Consequently, the computational savings due to hierarchical localization are close to those ratios obtained by the topological level in chapter four.

Table 6.2. Metric localization performance in Heidelberg office environment– performance index versus method.

Performance index / **Method**	***Iterative Gauss-Newton method***	***Geometric Triangulation method***	***Geometric Triangulation method + FS***
Average positional x-error (cm)	6.1363	10.0167	4.3125
Average positional y-error (cm)	4.0432	34.4958	14.0962
Average rms error (cm)	7.3486	35.9207	14.7411
Average orientation error (°)	2.0789	5.0393	3.0052
Average execution time (msec)	69.842	22.008	-
Maximum execution time (msec)	149.366	23.974	-

6.6.3 Localization Evaluation using the Reference System

A more precise evaluation for the metric localization is conducted using the reference system that has been described in 6.6.1. The experimentation was carried out in the robotics laboratory of Heidelberg University. The lab comprises an open area of about 100 m^2. The Krypton camera is hanged on the ceiling, yielding a diagonal view for the test field, which restricts the navigation space to 2.0x1.8 m^2. Due to the constrained navigational space and the open-area structure of the lab, topological nodes are defined as six arbitrary views, as seen from the defined workspace. Three markers are mounted on the robot's platform in order to track the real pose of robot and generate the ground truth.

6.6.3.1. The Metric Map

The metric map in the new environment is generated in two exploratory steps. In the first exploration step, SIFT features are extracted for every topological node, and the proposed information-theoretic map generation is applied to extract the most relevant features. The extracted relevant features have two forms; the entropy-based feature form and the compressed codewords form. The compressed form represents the topological data part of the hybrid map, while the entropy-based form will be inputs to a second processing before generating the metric part of the hybrid map.

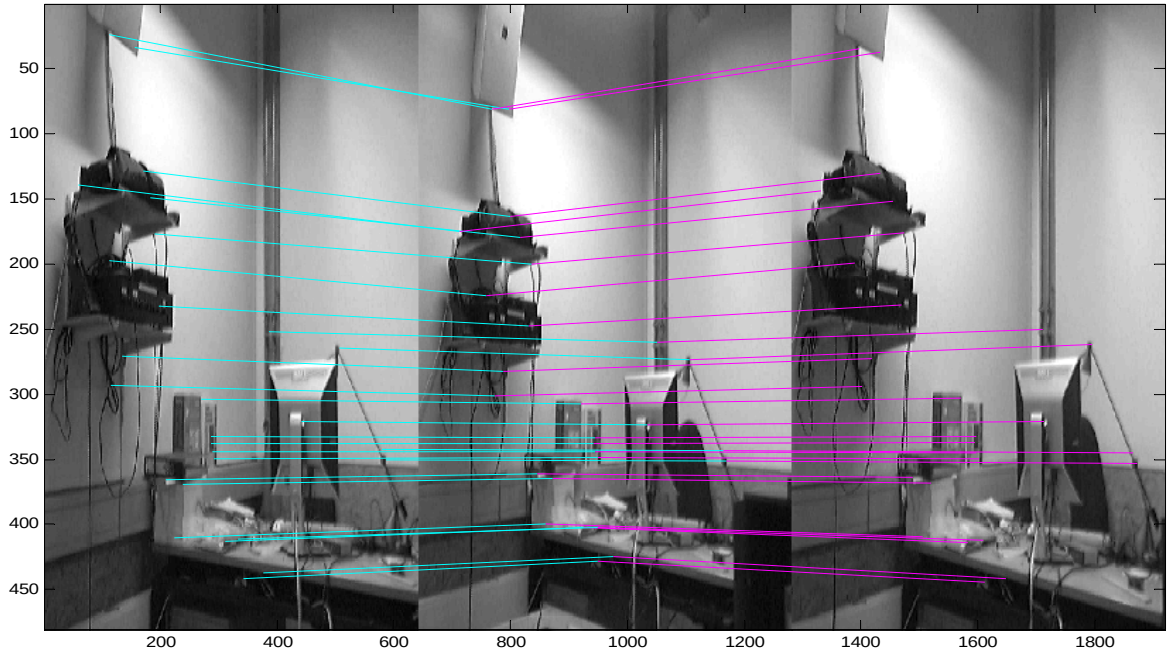
In the second exploration step, the features' metric information is calculated. To extract those positions while minimizing the errors, the reference system has been used to get robot poses and use them in a triangulation framework. The idea behind including robot poses measured by the reference system is to evaluate the localization, without the influence of

propagated metric map errors. Therefore, three robot poses are measured and their corresponding camera images are matched against the entropy-based features obtained in the previous step. Common features are next detected. Using the triangulation solution (photogrammetric model and the Newton-Gauss method), the 3-D positions of the commonly detected features are determined. Equations (6.18a-b) are solved for the position of features, together with the camera focal length as a parameter to be estimated. The solution to the metric map building (i.e. feature localization) is much easier than the robot localization's, because the non linearity of the Euler angles does not exist. The final hybrid map is now constructed as consisting of the topological codewords and the metric entropy-based features with their position information.

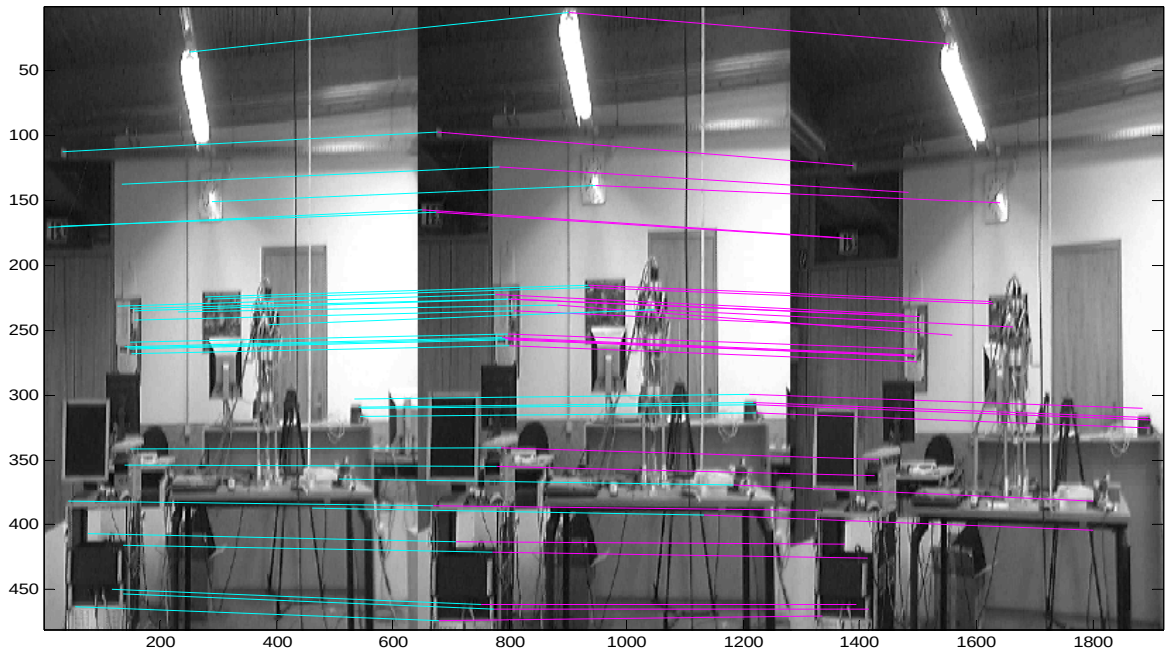
Figure 6.10 shows the process of feature matching for three camera views example, in which features are correctly matched, and subsequently localized. Figure 6.11 shows 3 and 2-dimensional view for the generated metric data obtained at the six topological nodes.

6.6.3.2. Metric Localization Evaluation

In these evaluating experiments, the robot is commanded to move a distance ahead of 2 meters from an arbitrary starting point in the navigating field, and then stop. Meanwhile, the real robot position (*the ground truth*) is determined by the reference system through the measured markers' positions. Simultaneously, the vision-based hierarchical localization is run on the laptop attached to the robot. Data from both steps are generated with timestamps for synchronization. As previously mentioned, hierarchical localization starts with fast topological matching, in which features are extracted for the current scene, compared to the codewords, and a majority voting signals the best node match. To extend the metric localization, the node's extracted features are matched to the entropy-based features to determine their identity, and the position of each feature is retrieved from the map. Finally, the filtering and selection criteria and the triangulation algorithm are executed. In the iterative Gauss-Newton method, the initial solution guess is assigned a default equal to the origin of the navigation field. It is noted that only the horizontal bearings are considered for the localization solution. This is chosen such that the vision sensor does not appear advantageous over other possible bearing-measuring sensors. Since the main purpose of the evaluation experiments is to quantify metric localization errors, quantified topological performance is not conducted in details in this section, as has been done in chapters four and five.



(a)



(b)

Figure 6.10. Three different view matching at three given robot poses for 3-D feature localization. Examples are for images acquired at topological nodes (a) Node1 and (b) Node4.

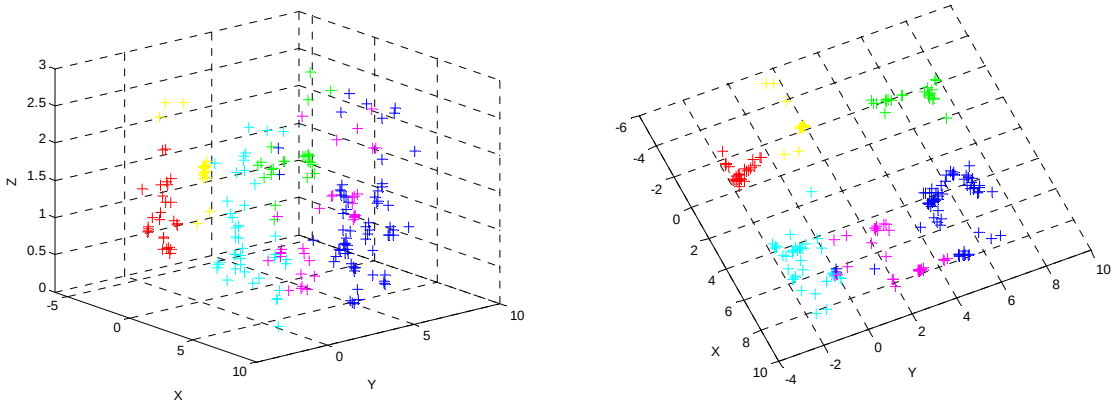


Figure 6.11. Three and two-dimensional view for the metric data.

Figures 6.12 to 6.18 show similar experiments conducted for the evaluation. In this experiment set, the iterative triangulation method is tested. The performance is evaluated in different nodes by comparing it to the ground truth. In all the cases tested, the topological place has been correctly identified by the robot at the beginning and all over the trajectory. Localization is executed similar to section 6.6.2. A topological place is first identified and the matched metric features are preprocessed using (6.52) to detect and exclude mismatched feature pairs and dynamic features. Next, triangulation is executed and the robot pose is estimated. Figures 6.12-a to 6.18-a show the 2-meter straight line ground truth trajectory driven by the robot which appear in blue color.

The example of figure 6.12-a is a localization experiment in node number one (Node1). The figure compares the ground truth in blue with the triangulation estimation in red. Figure 6.12-b shows the robot's trajectory with respect to the surrounding environment features. The positions of the total feature space are highlighted as green squares. The feature set observed by the robot during motion are marked as red squares. Those red squares are the features used during the entire movement, but they do not have to be used in every single position estimate calculation. Figure 6.12-c shows the equivalent orientation error during the experiment. The average localization error in the pose vector is (13.9, 14.7, 8.9)[4] with a maximum value of (22.3, 35.6, 9). This experiment localizes the robot at a starting distance of 4.4 meters from the furthest detected feature.

[4] Units are (centimetres, centimetres, degrees) for the robot position vector.

Figure 6.13 shows a second experiment also conducted in Node1. This time, the robot has a better distribution for the observed features in the camera view (i.e. the features are spread on both sides of the robot). This generates less localization errors in both the robot's position and orientation. The average localization error in the pose vector is (4.4, 6.8, 10) with a maximum value of (17.8, 20.8, 10.5).

Figure 6.14 shows another experiment conducted in node number four (Node4), where the identified features are much further. The furthest detected feature is at a distance of 6 meters. The topological node has been correctly identified, with less localization errors identified because of the better feature distribution in the robot's view. The average localization error in the pose vector is (6.4, 5.9, 3.1) with a maximum value of (9.9, 9.5, 3.4). Figure 6.15 shows an experiment conducted in the same node again. This time, the Gauss-Newton method is assigned an initial pose value which is far away from truth. The position estimates are correctly identified but the orientation estimate is not precisely identified. It is shifted by 180 degrees. The average localization error in the pose vector is (2.1, 6.8, 176.5) with a maximum value of (2.7, 15.6, 177).

Figure 6.16 shows an experiment in topological node number six (Node6). The triangulation could not detect the features at the beginning and at the end of the trajectory because the features are grouped on one side and are sometimes undetected. The average localization error in the pose vector is (4.3, 8.1, 5.2) with a maximum value of (16.6, 21.1, 6.4). Figure 6.17 shows an experiment conducted in topological node number five (Node5). The experiment has been carried out with two different initializing estimates. The first one managed to converge to the right orientation, while the other failed, and a shift of 180 degrees occurred. The average localization error in the pose vector for the latter is (6, 2.4, 183) with a maximum value of (12.4, 7.7, 183).

Figure 6.18 shows an experiment in which the robot starts in Node1 and traverses a certain trajectory to reach a goal set in Node 4. During the turn, one of the three mounted markers is obscured by the robot's camera, which accounts for non-correct values in the ground truth as seen in the figure. Both topological nodes have been correctly identified. The orientation has been wrongly estimated in Node1, while correctly estimated in Node4 as shown in figure 6.17-c where a 180° shift exists. The average localization error in the pose vector is (1.5, 11.1, 89) with a maximum value of (3.6, 17.6, 179).

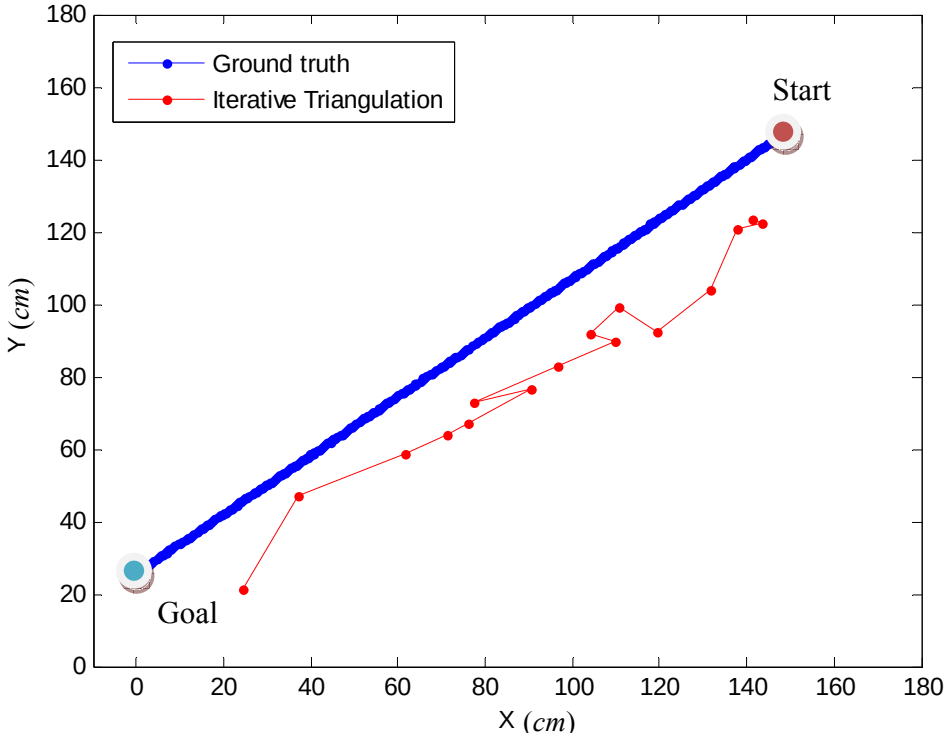


Figure 6.12. (a) Metric localization performance in topological node one (Node1).

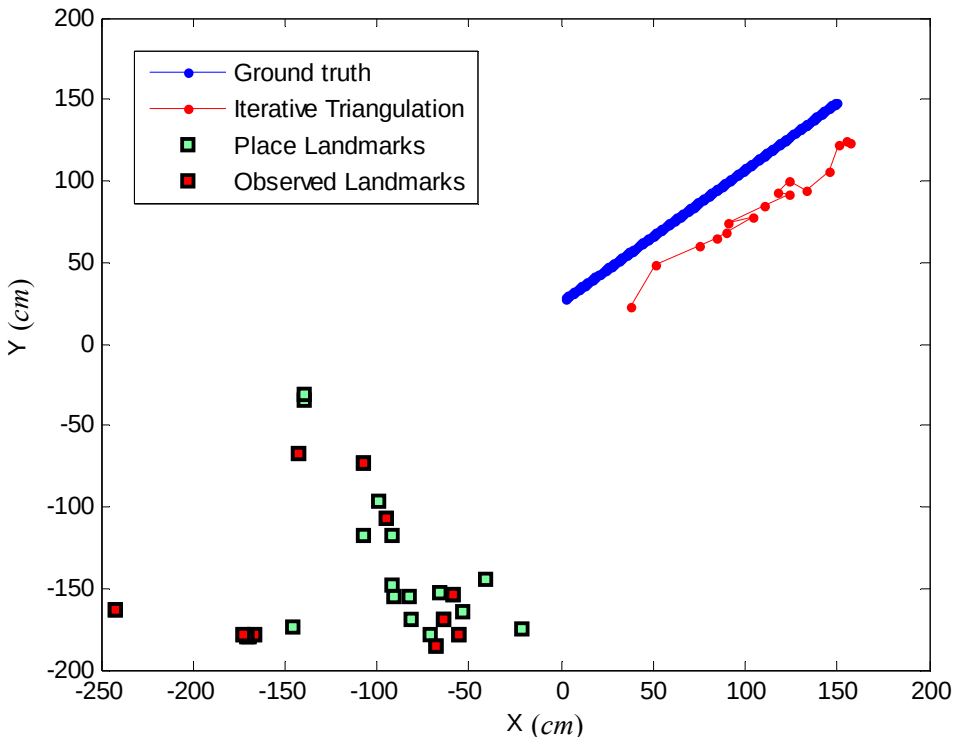


Figure 6.12. (b) Metric localization performance in topological node one (Node1).

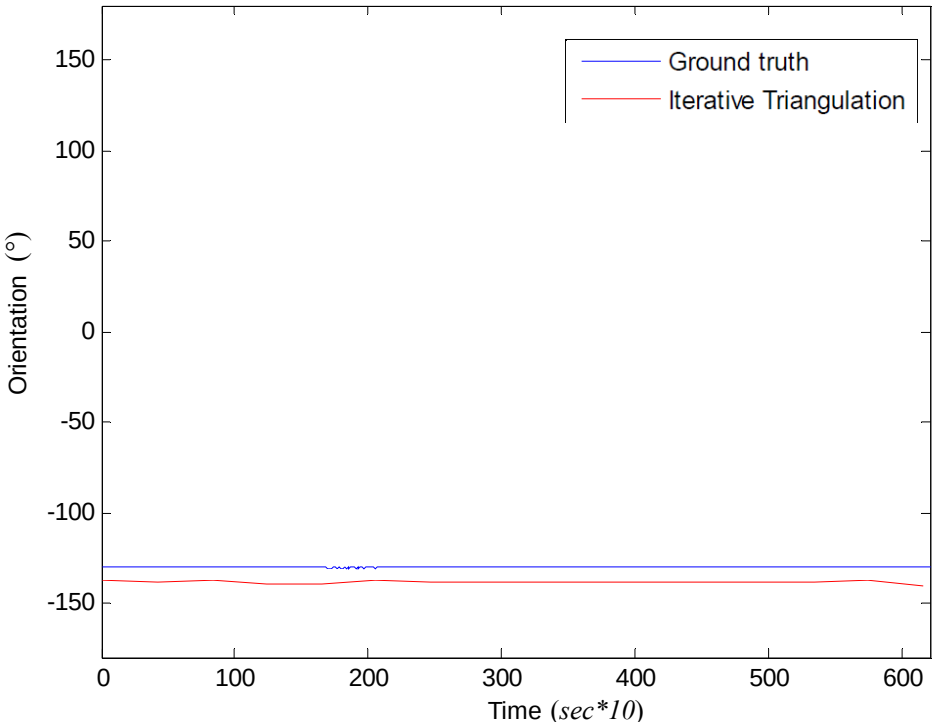


Figure 6.12. (c) Metric localization performance in topological node one (Node1).

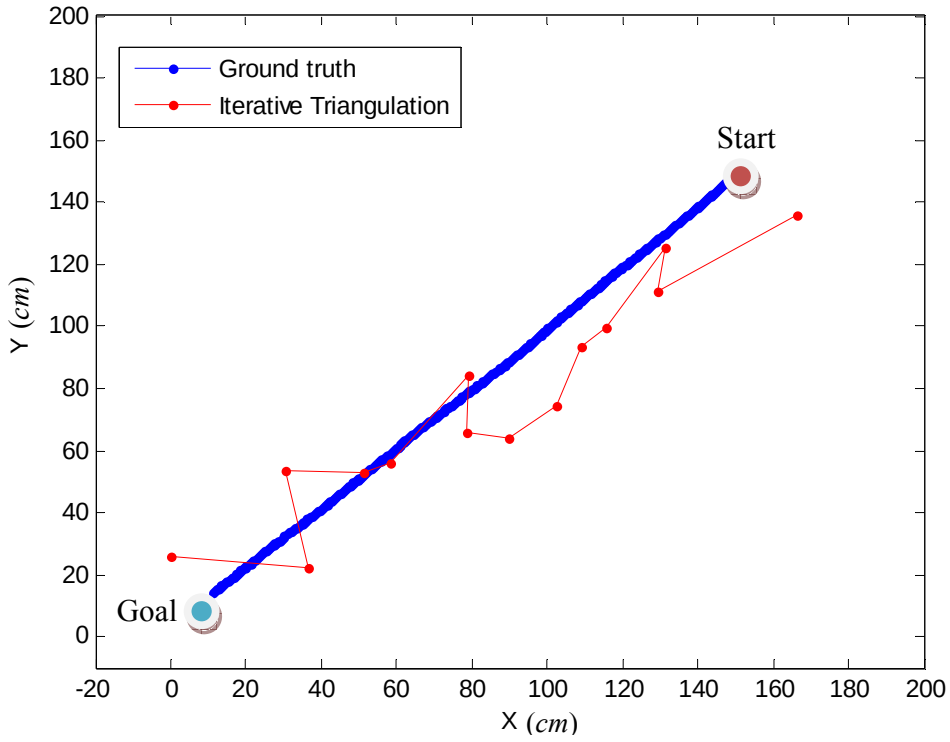


Figure 6.13. (a) Metric localization performance in topological node one (Node1).

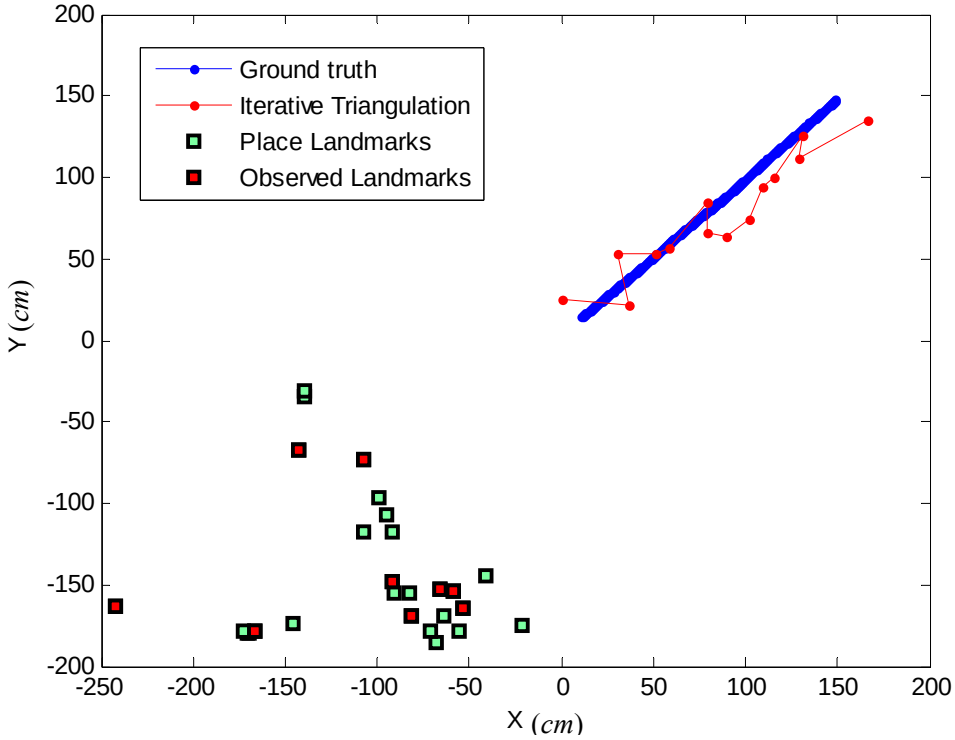


Figure 6.13. (b) Metric localization performance in topological node one (Node1).

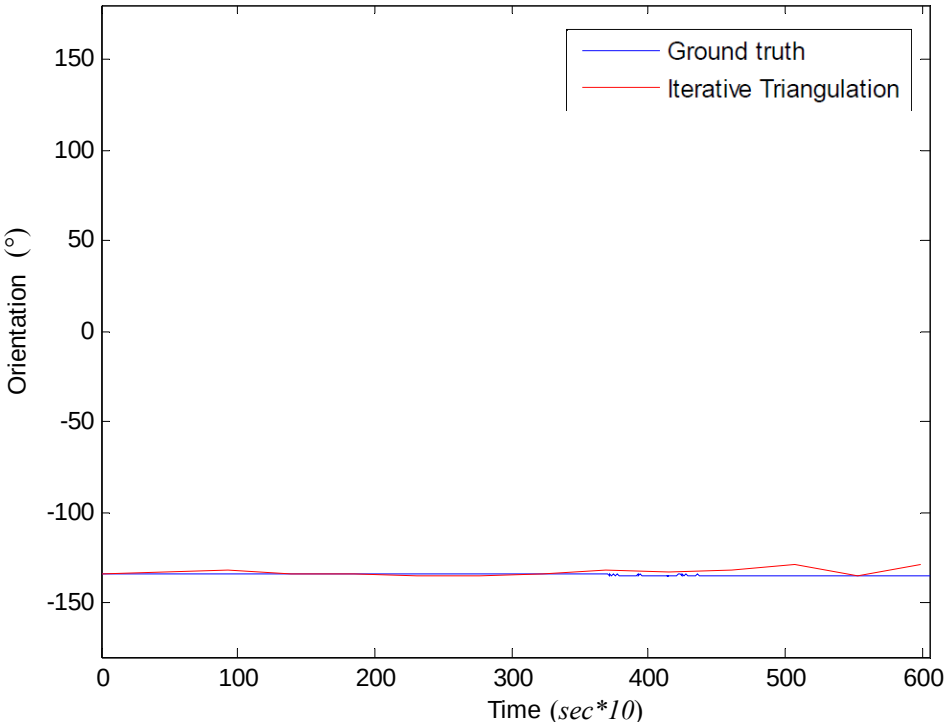


Figure 6.13. (c) Metric localization performance in topological node four (Node4).

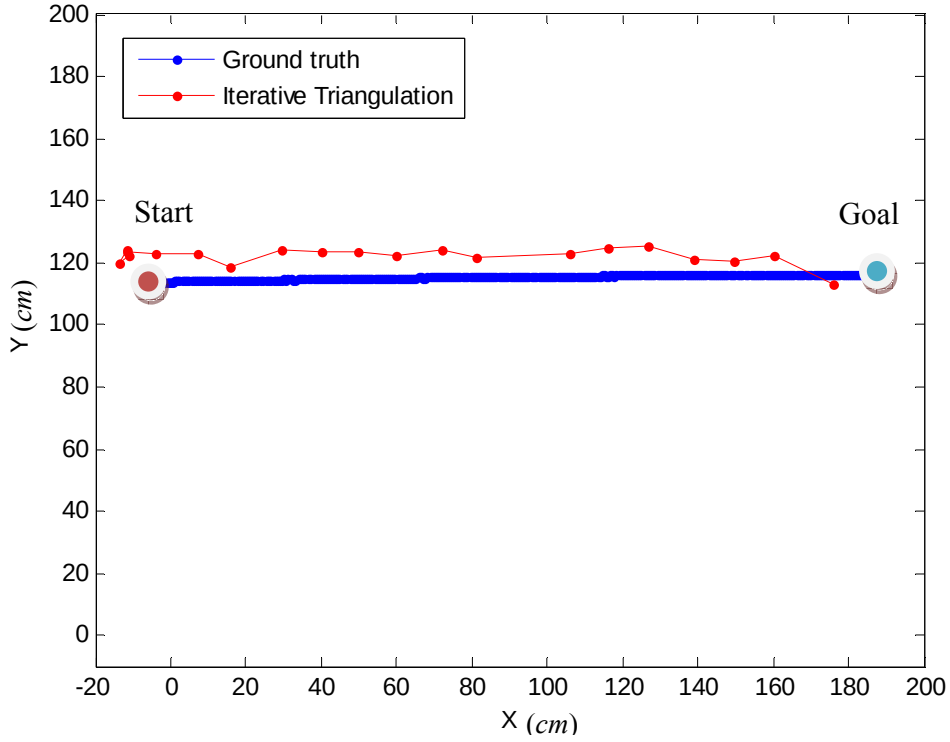


Figure 6.14. (a) Metric localization performance in topological node four (Node4).

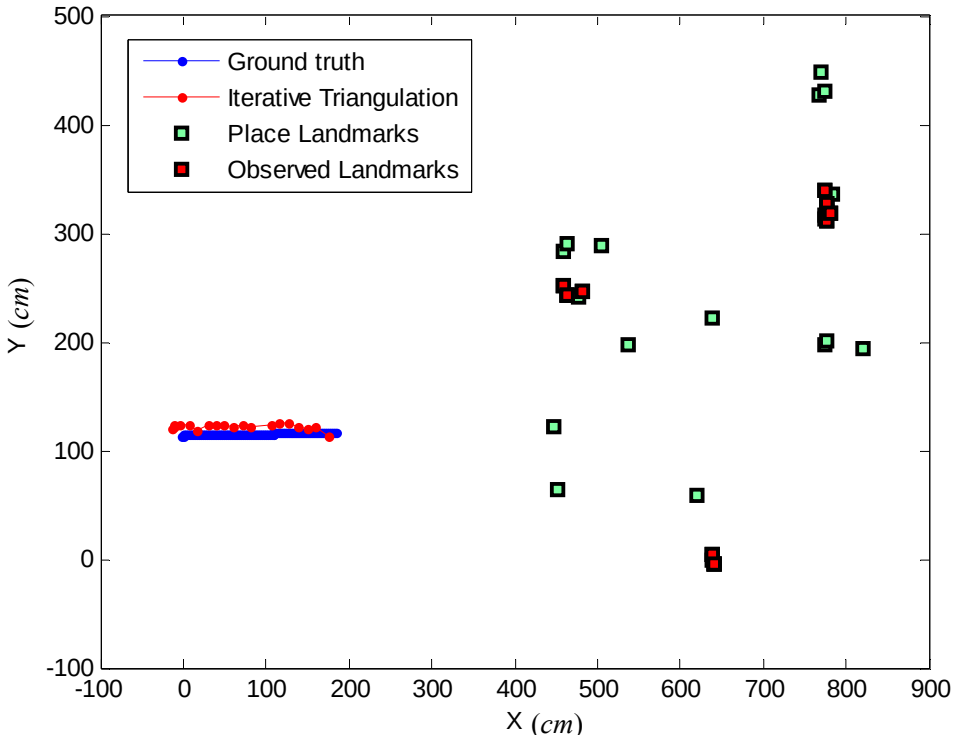


Figure 6.14. (b) Metric localization performance in topological node four (Node4).

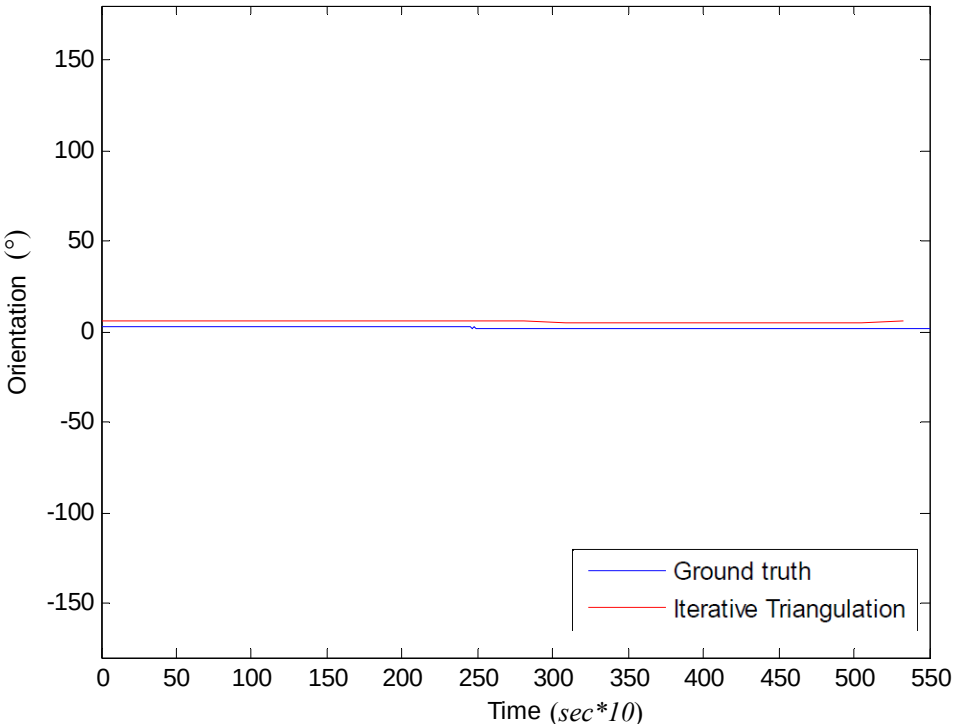


Figure 6.14. (c) Metric localization performance in topological node four (Node4).

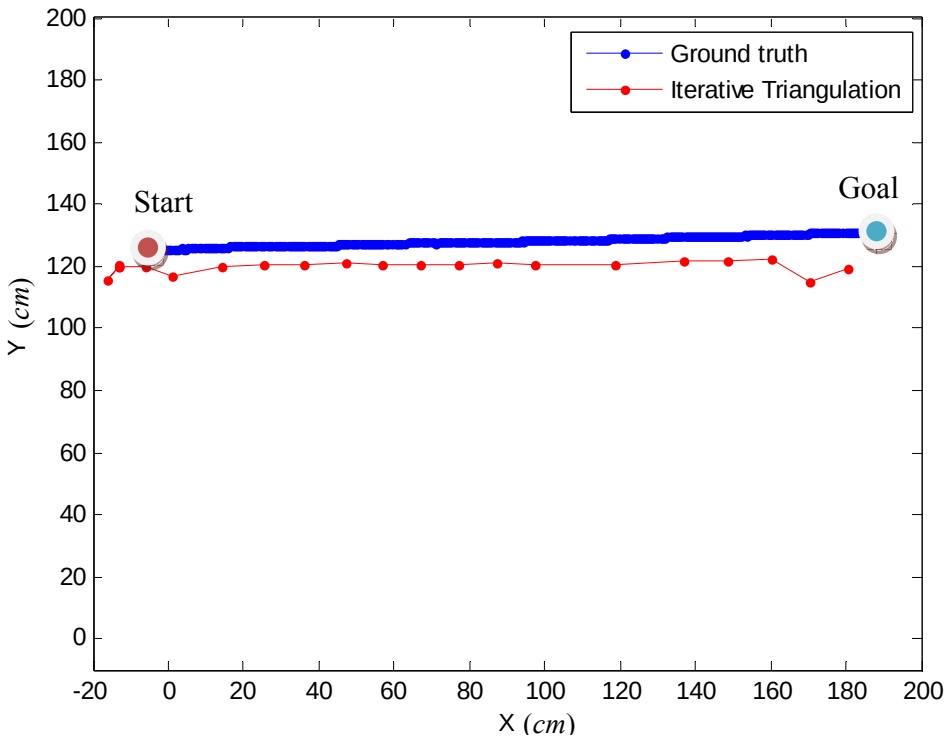


Figure 6.15. (a) Metric localization performance in topological node four (Node4).

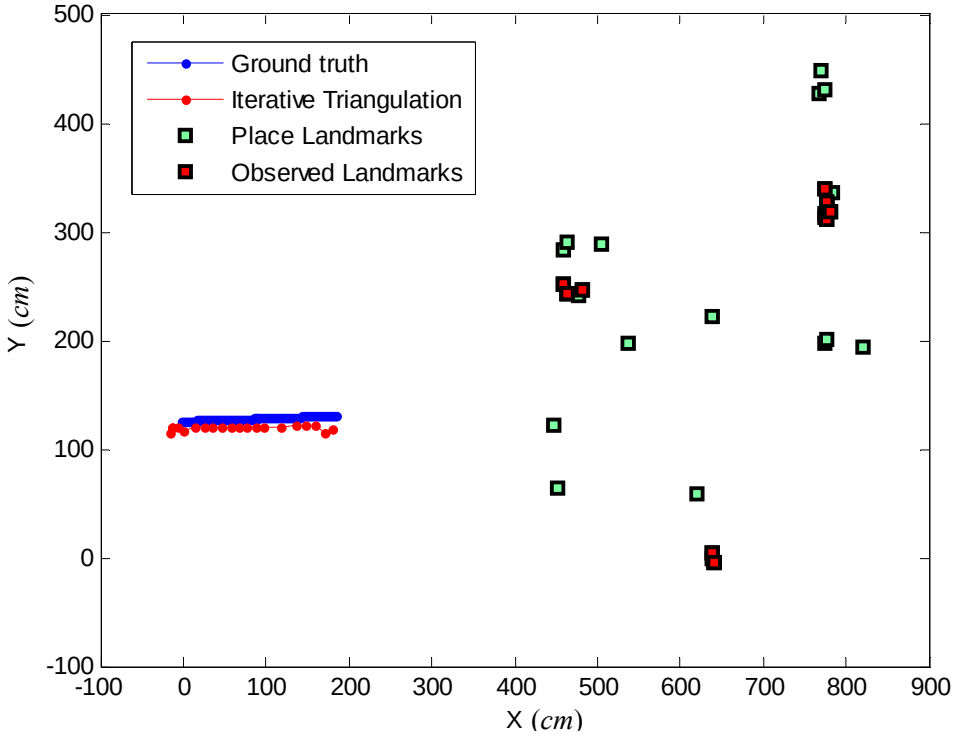


Figure 6.15. (b) Metric localization performance in topological node four (Node4).

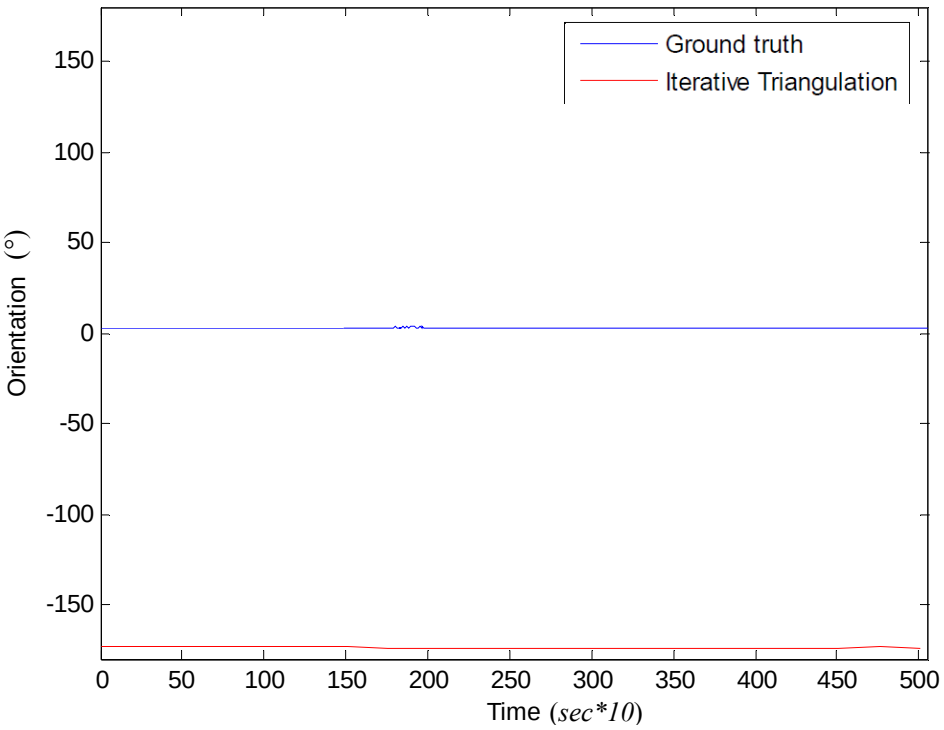


Figure 6.15. (c) Metric localization performance in topological node four (Node4).

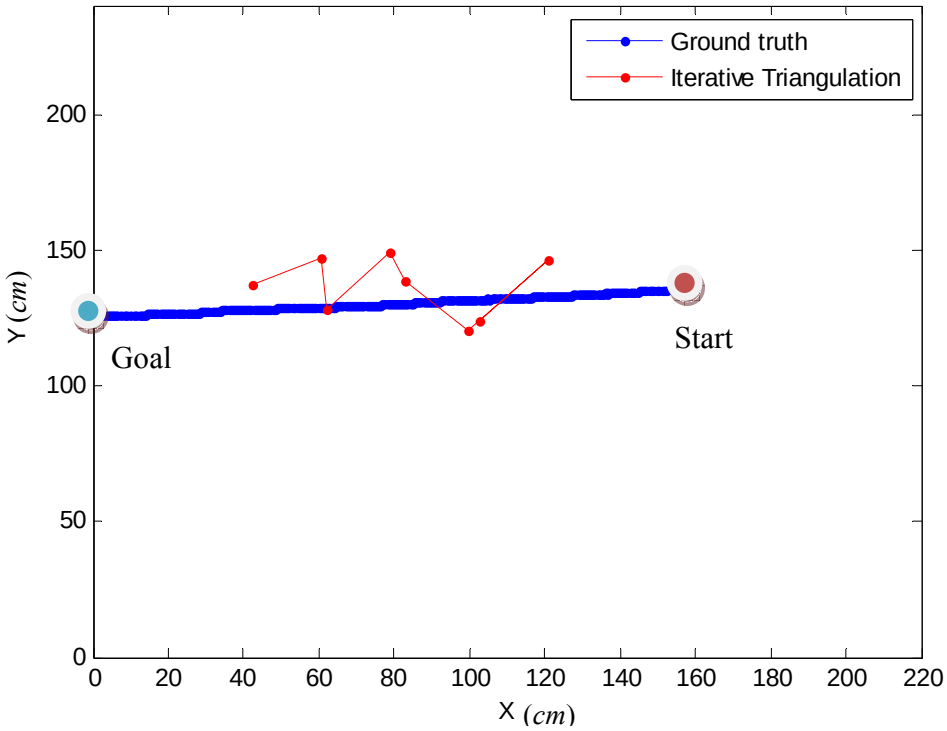


Figure 6.16. (a) Metric localization performance in topological node six (Node6).

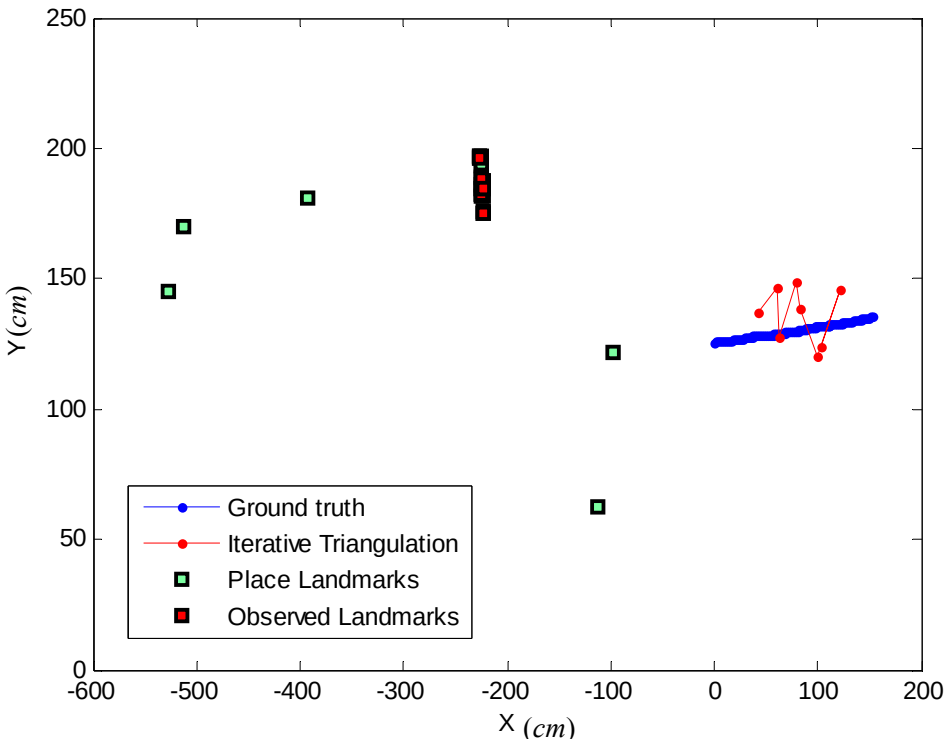


Figure 6.16. (b) Metric localization performance in topological node six (Node6).

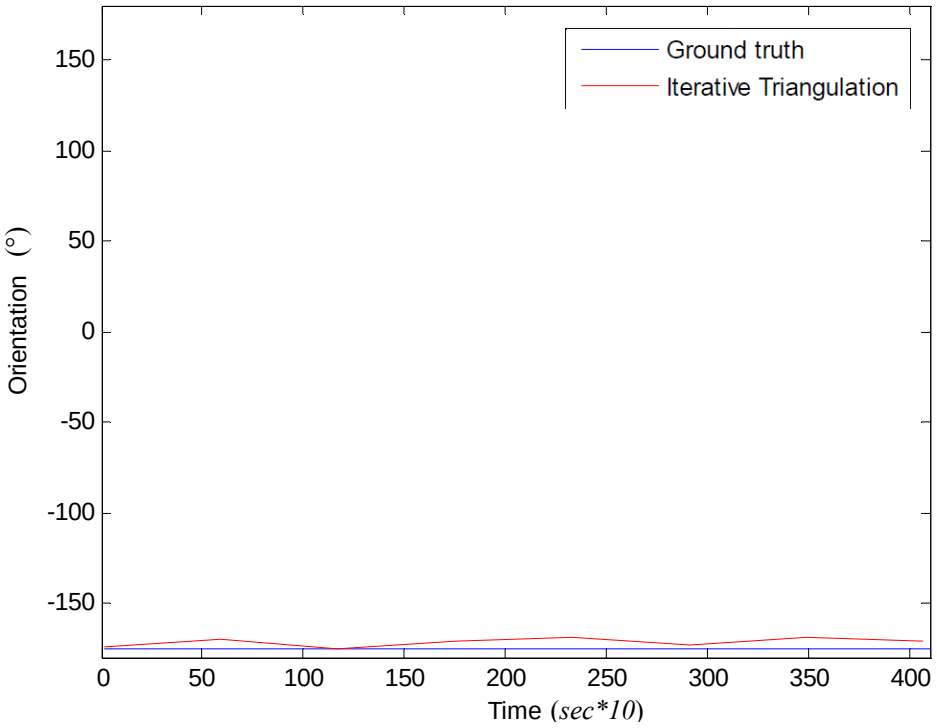


Figure 6.16. (c) Metric localization performance in topological node six (Node6).

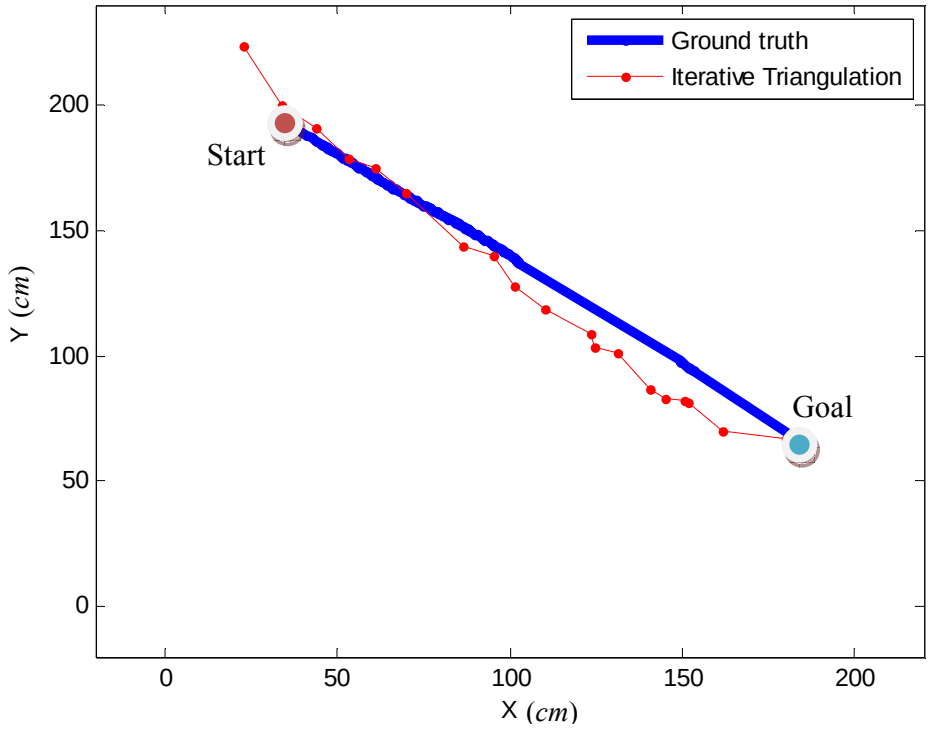


Figure 6.17. (a) Metric localization performance in topological node five (Node5).

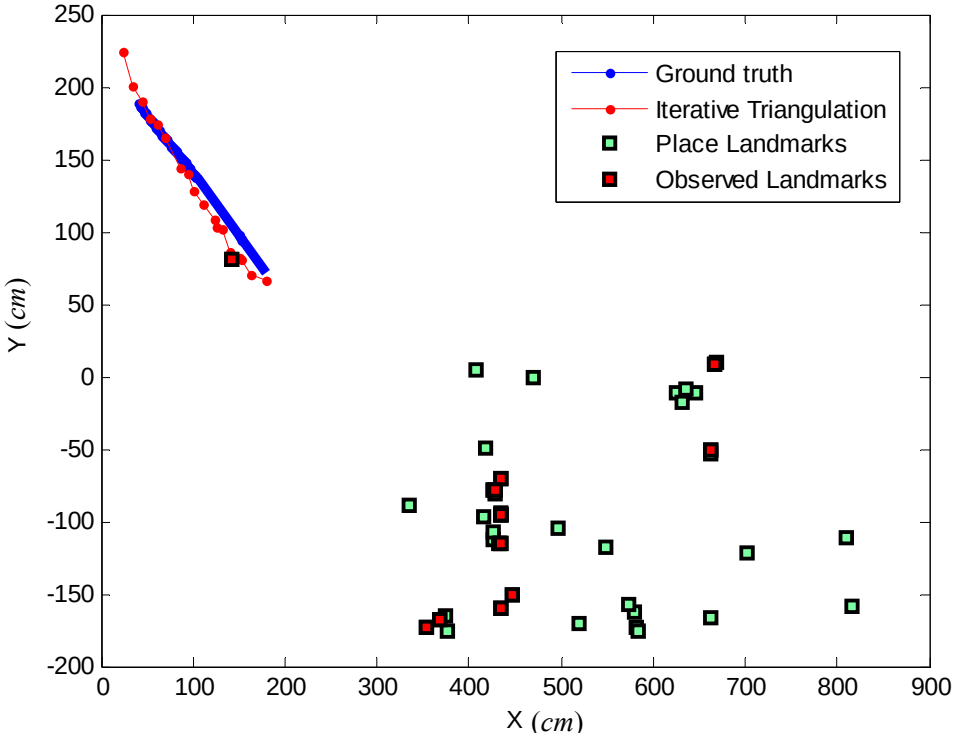


Figure 6.17. (b) Metric localization performance in topological node five (Node5).

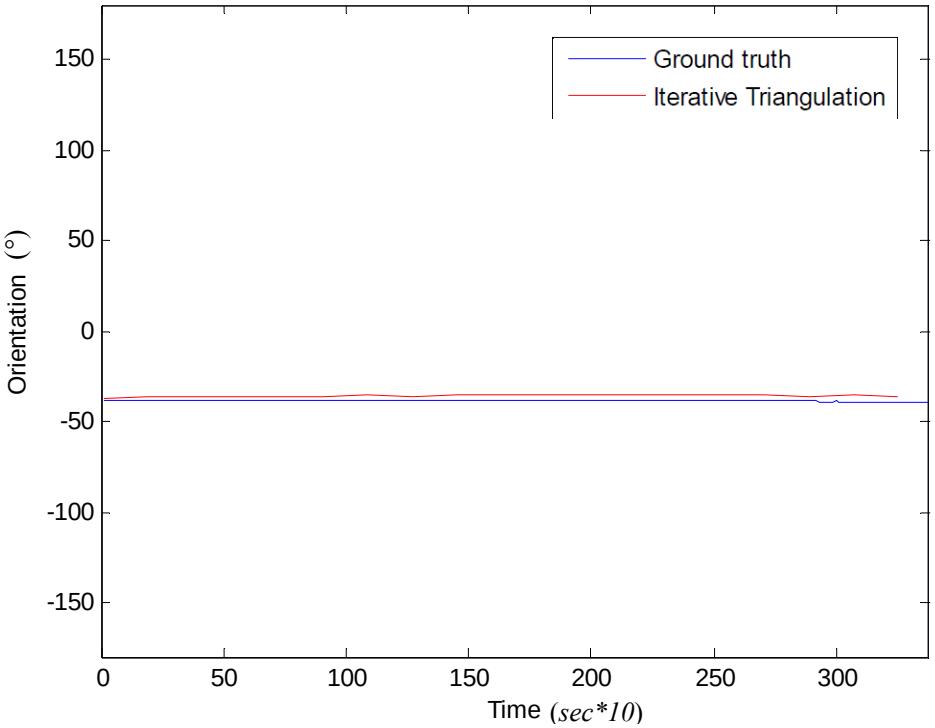


Figure 6.17. (c) Metric localization performance in topological node five (Node5).

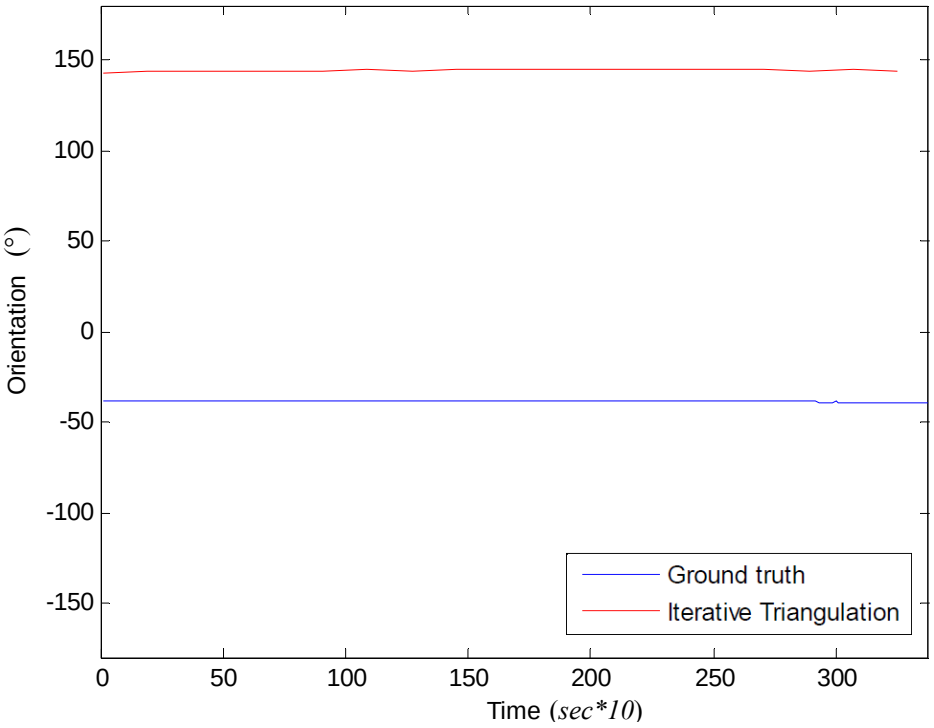


Figure 6.17. (d) Metric localization performance in topological node five (Node5).

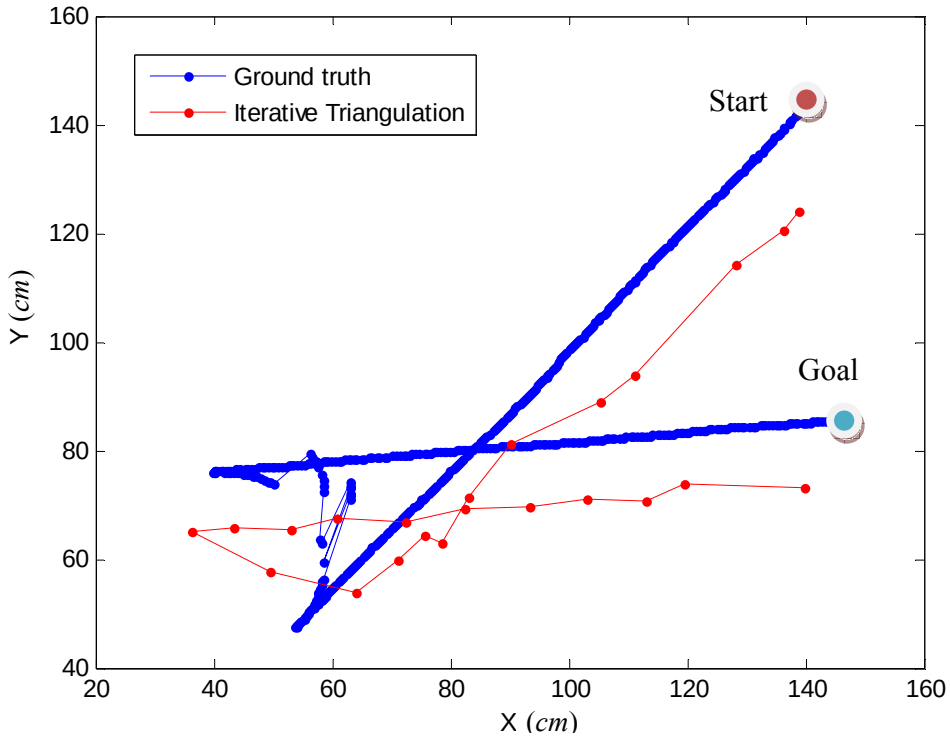


Figure 6.18. (a) Metric localization performance starting in Node1 and switching to Node4.

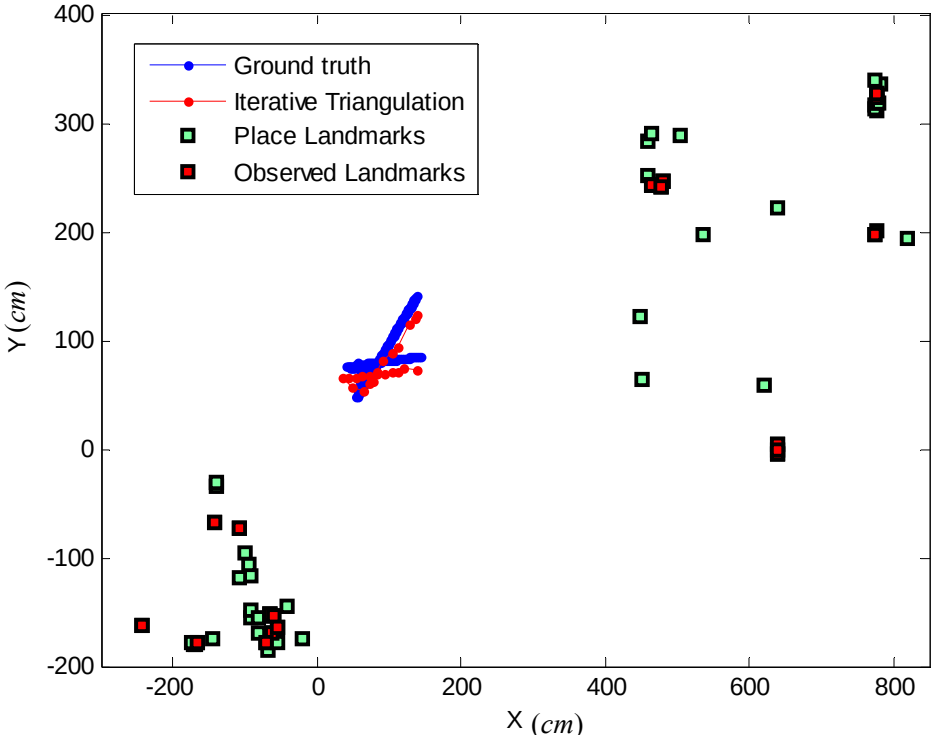


Figure 6.18. (b) Metric localization performance starting in Node1 and switching to Node4.

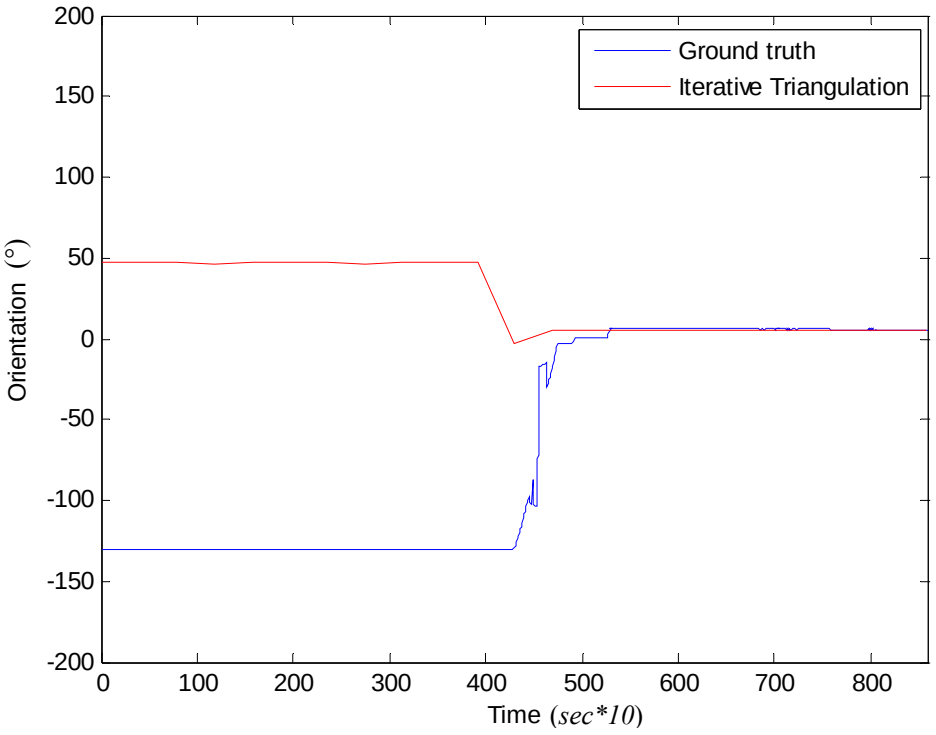


Figure 6.18. (c) Metric localization performance starting in Node1 and switching to Node4.

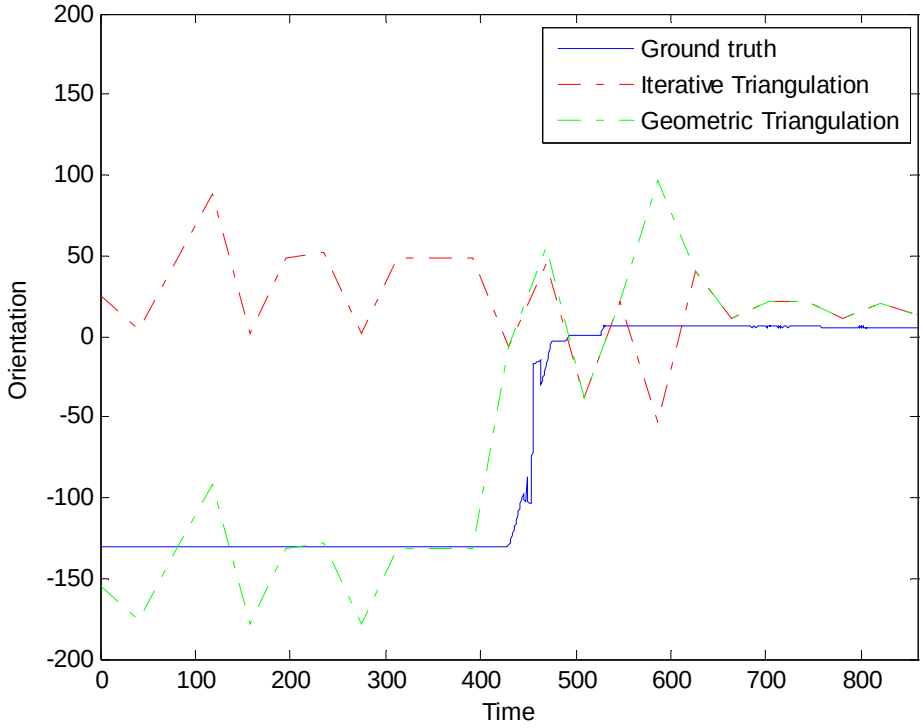


Figure 6.18. (d) Metric localization performance starting in Node1 and switching to Node4.

Figure 6.18-d compares the orientation calculated by the Gauss-Newton method to that of the Geometric Triangulation for the trajectory of figure 6.18-a when using 3 landmarks only. The figure shows that the solution by the Geometric Triangulation method gives a more robust (i.e. non-shifted) estimate as compared to the solution by the Gauss-Newton method.

Figure 6.19 shows an experiment example for comparing the robot pose vector using both the iterative and Geometric triangulation methods. The iterative method uses multiple landmarks (*>3*), while the Geometric method uses three landmarks only. From one side, it is obvious that more landmarks triangulation acquires more precise estimate. That is why errors exhibited in the Geometric Triangulation estimates are higher. However, from another side, the Geometric method preserves robustness for the orientation estimation as indicated in figure 6.19-b, which is missed in the iterative solution. For this reason, the Geometric Triangulation can be used to guide the iterative method for correct initialization values. It is noted that the process of landmark detection from image processing often includes poor measurement accuracy, because of the sensor characteristics and noisy measurement values. Using more than three landmarks can be efficient to increase such accuracy.

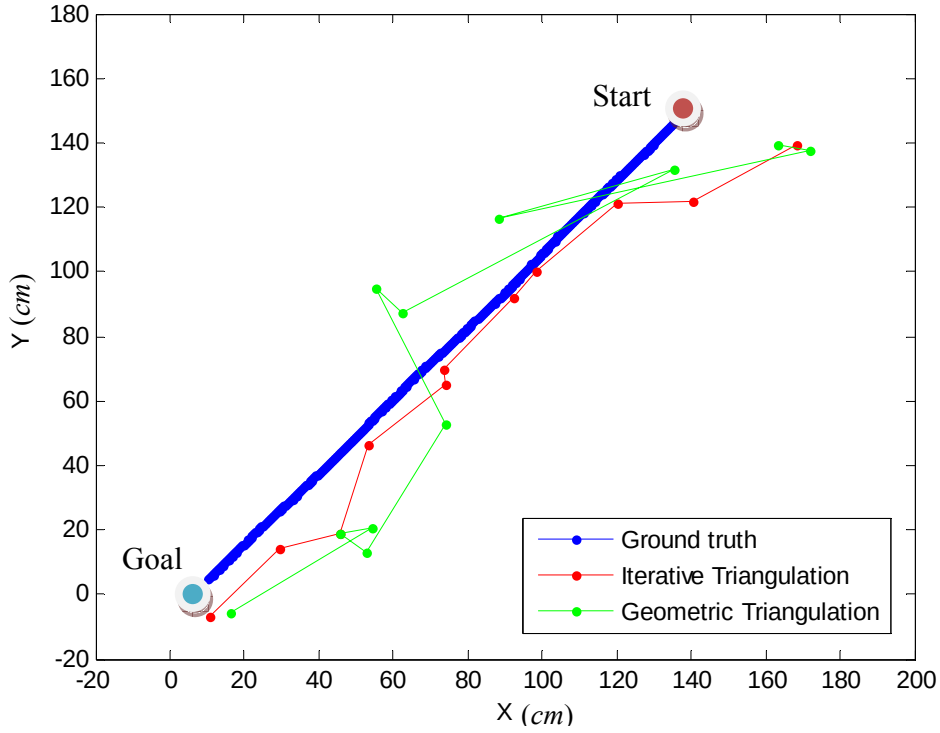


Figure 6.19. (a) Metric localization performance – Redundant features performance.

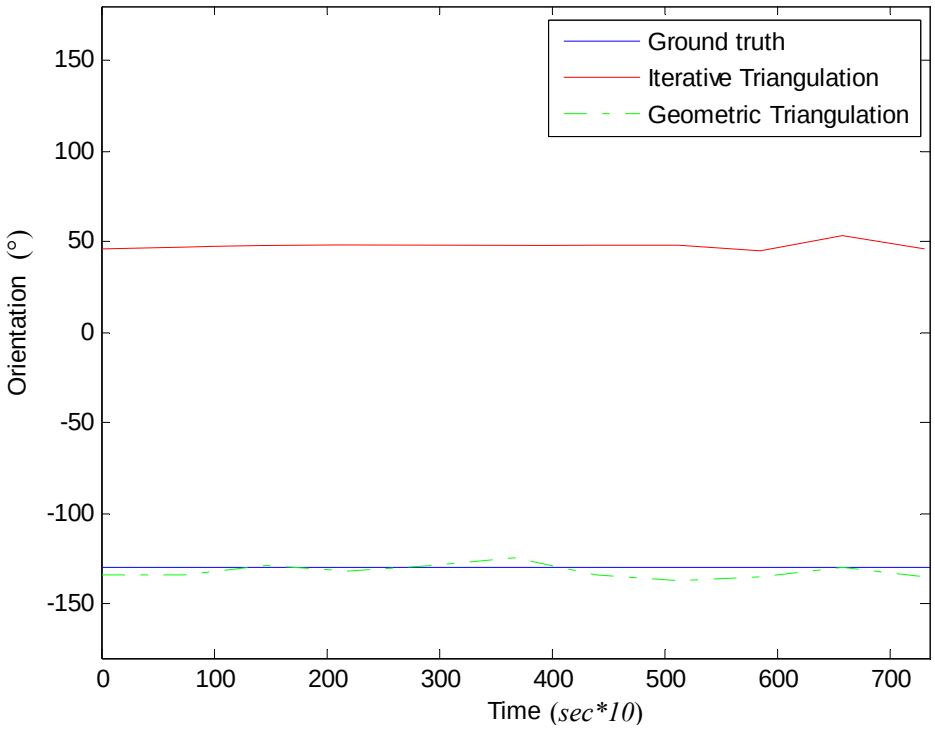


Figure 6.19. (b) Metric localization performance – Redundant features performance.

Figure 6.20 shows an experiment example to illustrate the effect of employing the metric feature selection criterion introduced in 6.5.2. The best 3-landmark configuration has been selected using equation (6.54) that combines measures of dispersion and non-collinearity. The figure compares the Geometric Triangulation method in two cases: a case of random selection for any three visible features versus another case where three visible features are selected by the criterion. The proposed selection criterion provided more accurate position estimation (less localization errors) than the random selection for most of the trajectory. The figure also compares the Geometric Triangulation to the iterative triangulation that uses several features. The experiment again ensures that triangulating several features minimizes the localization errors.

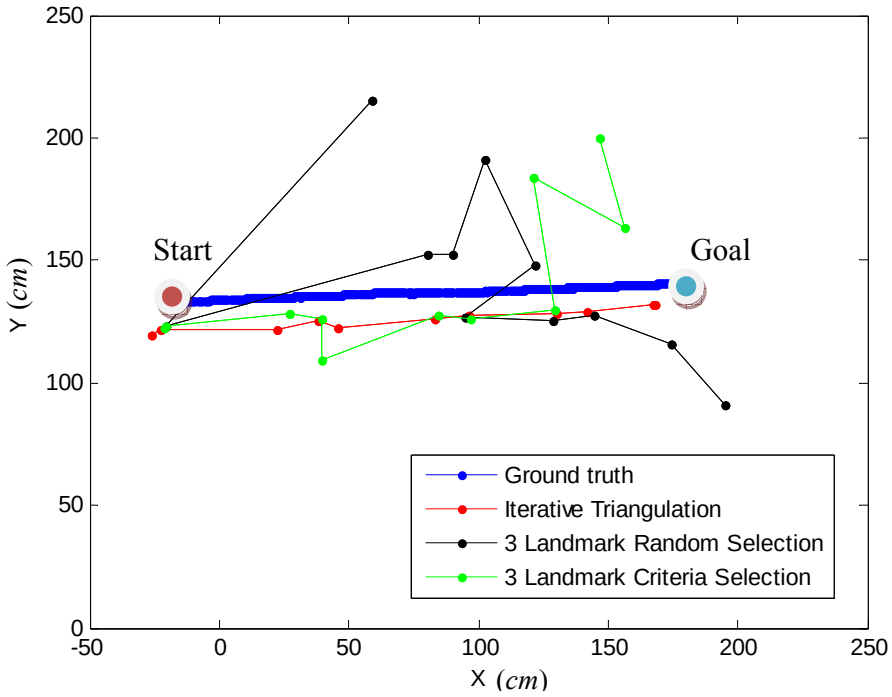


Figure 6.20. Metric localization performance – Criteria employment performance.

After conducting several experiments, the average localization errors have been recorded and summarized in table 6.3. The average x and y errors are about 4 centimeters when using the iterative triangulation method. The average root mean square error (rms) is about 16 centimeters. The average orientation error is on average 6° (excluding the 180° orientation shifts cases). The detailed performance of the iterative triangulation method, Geometric

Triangulation with feature selection criterion, as well as Geometric Triangulation with random feature selection is summarized in the table. The Geometric triangulation exhibited higher localization errors than the iterative Gauss-Newton since it uses three landmarks only in the solution. Employing the criterion of feature selection shows less localization error than the random feature selection.

Apart from the recorded results of the implementations, some sources introduced errors in the estimations, which can be handled. One source is the metric map building process. The position data in the feature-map have a few-centimeter error bound (about 2-3 centimeters). This error bound has been validated by single point measurements using the Spaceprobe of reference system. Triangulation with several robot poses in the metric map building, instead of the three poses applied in our case, helps in minimizing those errors. Another error source is the used camera because it is not optimally tuned with respect to its intrinsic parameters (e.g. lens distortion has been ignored). A final source of error originates from the camera setup. The camera has been fixed on a metal support to provide suitable height to capture the environment features. This, however, proved to be a source of vibrations that affected the bearing measurements and subsequently introduced measurement errors.

Table 6.3. Metric localization performance in Heidelberg robotics laboratory environment–performance index versus method.

Performance index / **Method**	***Iterative Gauss-Newton method***	***Geometric Triangulation method***	***Geometric Triangulation method + FS***
Average positional x-error (cm)	4.3775	27.0891	14.0323
Average positional y-error (cm)	4.7784	28.5386	18.5011
Average orientation error (°)	6.4	8.1	6.9
Maximum positional x-error (cm)	17.7998	74.9801	36.5800
Maximum positional y-error (cm)	25.8619	85.4723	58.0114
Maximum orientation error (°)	10.5	41.2	25
Average execution time (msec)	31.462	15.608	-
Average rms error (cm)	16.4030	51.0987	30.0826

Regarding the solution to the triangulation problem, direct or closed form solutions (e.g. Geometric Triangulation) frequently take less computational time than an iterative approach for the same positional accuracy. However, for real-time control problems, where

successive solutions should be produced at a high update rate, iterative solutions become highly efficient because previous estimates provide good starting point for the next iterations. Usually a single iteration is enough at each update period. However, an iterative method is sensitive to initial condition assignment, as has been encountered with the orientation estimation when it was assigned to a value far from the real one. In the general sense, this problem can be overcome through an estimate which can be obtained by sensor addition (e.g. a compass), combining other robust triangulation method (e.g. Geometric Triangulation), or using a pose tracking method.

It is worth mentioning some general notes about the information stored in the hybrid map, as well as the localization. First, the metric information of the features is stored relative to the topological node and not to the global environment space. This minimizes confusion with similar features distributed all over the environment. In other words, it avoids falling into the severe problems of data association and data correspondence. At the same time, it reduces the number of bits required for storing the metric information. Second, the metric localization provides feedback for maintaining the consistency of the hybrid map through the detection of mismatches and dynamic features. Such detection is tracked during the system operation on long terms, by preserving a variable along with each feature. Every time a mismatch or dynamics is encountered, the stored variable is incremented. The feature can be eliminated from the map later when the variable reaches a certain threshold if needed. This assists the robot system to keep only robust and efficient features in the environment map.

In conclusion, the advantage of the hierarchical solution presented in this chapter is mainly seen in the space and time complexities, which are significantly reduced to the dimension of the topological space (number of nodes). Compared to metric localization, which processes the entire space (accordingly entire feature space), the proposed hierarchical localization framework is computationally efficient by recording time and space savings between 60-90%. This has been verified in chapters four and five. The required accuracy of identifying a topological location by the first level of hierarchy is high, relying on the information-theoretic modeling for the environment. A final advantage for hierarchical frameworks is that they provide a means to employ and concurrently switch both arts of navigation, the topological and metric, as the robot is not required to work on the exhaustive metric level most of the time.

6.7 Summary

This chapter has presented a hierarchical localization framework suitable for dynamic large-scale environments. A hybrid map is employed for localization, which comprises data with two different resolutions obtained from two domains. The low-resolution data is compressed features characterized by non-geometric information, and is used for topological place recognition in the first layer of the hierarchy. Non-compressed form of the same features, but additionally complemented with metric clues accounts for high-resolution for the data. They are used for metric triangulation in the second layer of the hierarchy in order to estimate robot metric position. Since the topological level is modeled on information-theoretic basis, its accuracy is high enough to provide reliable hierarchical metric localization and robustness against place aliasing and misrecognition. Feature aliasing is also solved by detecting the environment dynamics as well as possible feature mismatches. This maintains accuracy and robustness of triangulation. Additional selection for the features based on their geometry and their dispersion in sensor view are proposed for augmenting more accurate pose estimates. The advantage of the proposed localization approach lies in the scalability and the lower complexity which has been revealed through the experimentations. Another significant advantage is that the approach does not rely at all on dead-reckoning information. Such issue provides an extensibility or fusion option with other localization approaches.

Chapter 7

Conclusion and Future Work

7.1 Summary

It is difficult to develop simple machine models that completely correlate with the conceptual view of similarity that the human brain uses. A human recognizes environment similarities and differences systematically, a matter that is decidedly difficult for a robot. From another perspective, it is difficult to develop a general model that suits every environment with the same efficiency for application. Therefore, current and future advances in robotic perceptual and identification systems invoke many challenges to be addressed, in order to construct reliable and information-rich models that can assist robot navigational tasks.

The work of this thesis targets an unattended research topic, which is the extraction of the minimal amount of information that supports efficient task execution. Such a topic is addressed for the benefit of environment model building and robot localization. Model building and localization are addressed in the context of large-scale, unstructured, densely cluttered and moderately dynamic environments. An information-theoretic-based solution approach is proposed, which employs structural components for generating information-rich maps and maximizing the localization accuracy while minimizing its complexity. The components provide solutions to the uncertainties caused by environment perceptual aliasing,

dynamics, and data correspondences, as well as to the space and time complexities caused by the high dimensions of features and space.

The thesis presents a hierarchical framework consisting of two levels. The framework incorporates both the topological and metric paradigms for map building and localization. This hierarchical framework maintains a single hybrid feature-map which is shared between the two levels. The localization module of the first level affords coarse place identification, while that of the second level provides a more precise geometric estimate, given a previously identified topological place by the first level.

A basic contribution in the proposed work is a novel approach for modeling the environment data in the hierarchy. The hierarchy adopts an appearance-based modeling approach. It is independent of metric clues of features at the first hierarchical level and relies only on their occurrences in certain topological places. A proposed information-theoretic evaluation component regards the environment features as properties emitting certain transmission values that contribute to place and feature uncertainty minimization. The transmission values are a measure for the features' power of discrimination to categorize among the defined places, and are measured through a proposed conditional entropy filtering criterion. The evaluation component works in coordination with a second compression component by employing clustering techniques. The clustering provides significant compression ratios for the evaluated features. Only highly transmitting features are selected, while the rest are filtered out. Hence, two forms of data are generated from the evaluation and compression components: the selected *entropy-based features* and the compressed *codewords*. They represent high and low-resolution versions of the processed features respectively. Non-geometric descriptors of the codewords are preserved as topological data to be used at the first topological level. Non-geometric descriptors of the entropy-based features and their additional tagged positions account for metric data to be used at the second metric level. Having the same feature set but in two different forms yields a unified hybrid map at the two levels, with features capable of resolving topological and geometric locations.

Localization at the first level is conducted as a place matching problem, in which the current scene viewed by the robot is compared to the codewords. A fast, accurate and robust (against dynamics) topological localization is performed. A geometric localization at the second level is executed based on the identified place, in which metric visible features are

triangulated. Relevance in the choice of features is made according to their structure (e.g. the more dispersed, the less collinear in real-world). Additionally, redundant feature triangulation is used to obtain a better estimate for the robot pose. Using the robot's spatial view, dynamic features and mismatches are detected and excluded to ensure robust and accurate triangulation. A feedback on those undesired features is reported by the metric level to consider whether they will be preserved in the hybrid map or not. It happens that many environment objects move around, for example in house or office environments, due to human activities. These dynamic objects are useful for the topological localization and can be consequently still preserved for the topological level. They are, however, not beneficial for the metric localization and should be excluded as metric features.

The proposed hierarchical framework augments accurate and less-complex localization, by utilizing only the data with the highest information content (as measured by their transmission rates) and the hierarchy concept. Additionally, the introduced information-theoretic modeling approach refrains from any specific environment characterization or object modeling, such as the commonly used geometric modeling approach. Relying on a hybrid quality-based map, small-size maps are generated. The information-theoretic evaluation component provides quantified values for the environment properties in terms of a transmission value. Therefore, a poor environment can be detected and hence augmented with external referencing aiding solutions as localization accuracy is expected to be low.

The proposed solutions have been demonstrated using a vision sensor, as being the most fundamental, important and rich information provider. Scale invariant local features, which are robust against several transformations and occlusions, are used. Employing the information-theoretic solution achieves a two-fold benefit by reducing the features' disadvantageous huge size and selecting the best discriminative set for a place at the same time. The topological solution has been tested thoroughly using two different datasets and two different vision sensors; perspective and omnidirectional. The metric solution has also been verified in two different operating environments. The combined hybrid solution proves to achieve a good combination of accuracy-space-time performances, and suiting large-scale, dynamic and unstructured operating environments.

The general advantages of the proposed information-theoretic solution approach and hierarchical framework can be summarized as follows: (1) The generality in being applied to

any environment, sensor and feature extraction methodology; (2) providing accurate multi-resolution localization with efficient resource management; (3) scalability to large-scale and complex environments; (4) robustness to dynamics of environment. On other technical implementation levels, the approach has the pros of: (1) Automatically selecting natural features and building an environment map without any environment modification; (2) minimizing the perceptual aliasing and data correspondence problems of common similar places and features; (3) providing a purely-vision localization solution with independence of dead reckoning information.

7.2 Scope and Limitations of Work

The proposed solution structure and methods adapt well to unstructured environments, where it is difficult to extract well-defined geometric features. In a similar way, it adapts well to densely cluttered environments, where objects can obscure each other, and hence important features are hidden. The approach is also practical and computationally efficient for implementation in large-scale environments due to the hierarchical structure and feature reduction techniques employed. It can provide substantial memory and computation savings. The issue might not be the storage itself, since large storage capacities are available at low cost. It is the searching and matching processes involving large data collections, which remain the main challenge for computer systems.

Regarding the approach's application, no restrictions on the environment are imposed, except for a plain background or an environment that lacks much detail. In those environments, the approach might perform poorly, and it may sound more logical to employ external referencing (e.g. artificial landmarks). The approach is also unsuitable for places that show very a high degree of similarity where unique features hardly exist (see for example figure 1.4 for a severe perceptual aliasing problem). In such case, the approach has to be integrated with a component that can resolve this severe ambiguity. A multimodal probability distribution over place hypotheses can be tracked using temporal evidence, and hence provides a possible solution to the problem (e.g. probabilistic filtering like Markov models or particle filters as pose tracking solution). This distribution generation option is already there in the topological localization module. As the robot moves through its environment, the probability distribution will be updated on the basis of the new sensory evidence and

observed motion of robot since its previous position. Accordingly, the place ambiguities are resolved. In this context, it is assured that involving a topological feature-map based on information richness and distinguishability will assist this suggested solution against the severe ambiguities of some environment places.

A property of the presented approach is that it is holistic. It is applied on the entire scene where natural features are acquired automatically without a need for prior feature or object specification. In an exact manner, the approach can be applied to characterize and recognize specific object classes (e.g. a wall painting, fire distinguisher or other objects), which can be deposited in environments with poor details. These object classes, which can be either artificial or natural, will act as landmarks to assist the local navigation.

The proposed approach is applicable for several autonomous mobile robot applications. Specific applications include service robots, such as museum-guide tours, where it can guide a human or a robot over large environments influenced by several dynamic variations. The approach can additionally provide a navigational aid for visually impaired people and wheelchair users. Furthermore, the stand-alone topological level can serve as a place recognition application for mobile devices.

7.3 Directions for Future Work

Future work suggests some enhancements for the presented methods, as well as other additional concepts to be integrated, which can enhance the general structure and performance.

Regarding the implemented methods, it is desirable to regard more technical details for the generation of visual codebooks. The sampling of the patches or clusters is the critical component in any BoW method, as well as the consideration of the codeword value in the cluster. Combining Gaussian operators in clustering, instead of sharp clustering, can decide for codewords that are critically stable. This factor, in addition to studying the cluster density distribution, can generate more efficient codewords. Furthermore, other clustering algorithms than the *k*-means (e.g. *k*-medoids), and other distances rather than the Euclidean, can also be investigated.

For the general solution structure, the use of fused data modalities from different sensors is encouraged. It is also suggested that an inspection of the approach in larger environments be carried out. More specifically to vision, implementing the local features algorithms using hardware can be regarded [Zhang et al., 2008; Kwon et al., 2008]. This issue is quite promising for real-time implementations. Testing the localization with metric map-building methods in [Jebara et al., 1999; Min et al., 2007; Torr, 2002] are also suggested.

Otherwise, new modules can be integrated into the solution structure to overcome the severe perceptual aliasing problems, as previously introduced. It is a fact that in large-scale spaces, the structure appears at a higher significance than the observations available at an instant. Therefore, two components can make relevance to the solution design: temporal evidence and adjacencies modeling. Integrating system dynamics in a hybrid fashion at the second level, by fusing current with pervious data measurements, will overcome severe ambiguities. Additionally, the environment model and the localization will be significantly improved by verifying possible transitions between places. For instance, the robot cannot jump suddenly from the office to the kitchen without passing through a corridor, or it cannot move from the elevator view on the first floor to the other one on the third floor without ascending two floors. During such motion, other distinguishing places or measurements will be encountered, and therefore, the solution structure will offer enhanced performance.

Appendix -A-
K-Means Clustering

Clustering is the process of grouping a set of physical or abstract objects into classes exhibiting similarity. The class, which is termed a cluster, is a collection of the data objects that are similar to one another within the same cluster, whereas dissimilar to the objects in other clusters. Every cluster can be treated collectively as one group, and so clustering may be considered a form of data compression. Clustering is the reverse process of classification, where the set of data is first partitioned into groups based on data similarity, and then labels are assigned to the relatively small number of groups. Consequently, it is a form of learning by observation, rather than learning by examples (i.e. unsupervised learning method).

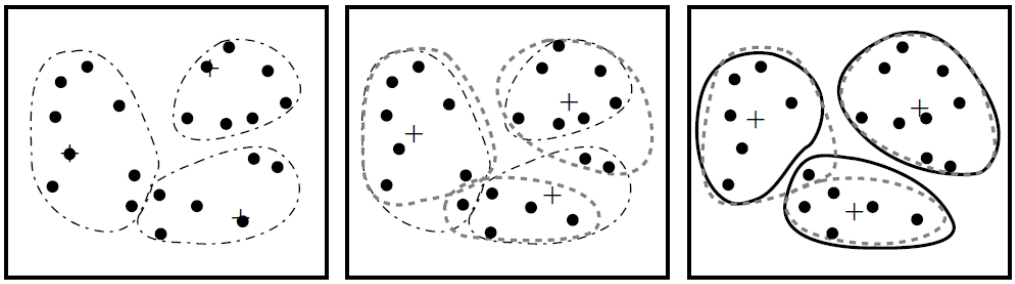
K-means [Han and Kamber, 2006] is one of the simplest algorithms in clustering analysis, which is classified as a partitioning method. It has been used in several problem domains. The algorithm classifies a given data set into a number of clusters (k), which is fixed a priori. The partitioning method creates an initial partitioning, and then uses an iterative relocation technique, which attempts to improve the partitioning by moving objects from one group to another. The criterion of obtaining good partitioning is that objects in the same cluster should be close or related to each other, whereas objects of different clusters should be far apart or very different.

The algorithm proceeds by defining k centroids (centers of gravity), one for each cluster. The centroids are initialized by picking k samples at random. The next step is to take each of the remaining objects in the given data set and link them to the nearest centroid. An object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean. When no point is pending, the first step is completed and an early groupage is completed. At this point, the prior assigned centroids are moved to the center of the clusters by re-calculating k new centroids. After having the k new clusters with k new centroids, a new binding is made between the data set points and the nearest new centroid as done before. The process is iterated in which the k centroids change their location step by step; until no more changes occur (i.e. centroids do not move any more). Hence, the centroids of the clusters are taken as the generate clusters. Figure A.1 shows a simple scenario for clustering into 3 groups, while figure A.2 summarizes the algorithm steps.

For the algorithm to converge, it sets up an *objective function* to be minimized. This objective function is a squared error function and has the formula:

$$E = \arg\max_{C} \sum_{i=1}^{k} \sum_{p \in C_i} \left\| p - m_i \right\|^2 \qquad \text{(A.1)}$$

where E is the sum of the square error for all objects in the data set; p is the point in space representing a given object; and m_i is the mean of cluster C_i (both p and m_i are multidimensional).That is to say, for each object in each cluster, the distance from the object to its cluster center is squared, and the distances are summed. This criterion attempts to make the resulting k clusters as compact and as separate as possible.



(a) (b) (c)

Figure A.1: The *k-means* partitioning algorithm

- Randomly initialize a set of K means $v_1^{(1)}, \ldots, v_K^{(1)}$
- Assignment step: assign each observation to the cluster with the closest mean.

$$E = \arg\max_{S} \sum_{i=1}^{K} \sum_{s_j \in S_i} \left\| s_j - v_i \right\|^2 \qquad \text{(A.1)}$$

- Update step: calculate the new means to be the centroid of the observations of the clusters.

$$S_i^{(t)} = \left\{ s_j : \left\| s_j - v_i^{(t)} \right\| \le \left\| s_j - v_{i^*}^{(t)} \right\|; \forall i^* = 1,\ldots,K \right\} \qquad \text{(A.2)}$$

$$v_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{s_j \in S_i^{(t)}} s_j \qquad \text{(A.3)}$$

Figure A.2: The *k*-means algorithm steps.

Although it can be proven that the procedure will always terminate, the *k*-means algorithm does not necessarily find the most optimal configuration corresponding to the global objective function minimum. The algorithm is sensitive to the initial randomly selected cluster centers. The algorithm can be run multiple times to reduce this effect.

In applying clustering approaches for data compression, the clusters are represented by the mean value of the objects in each cluster, such as in *k-means*, or by one of the objects located near the center of the cluster, such as in *k-medoids*.

Appendix -B-
Panorama Creation via Image Stitching

Panoramic imaging has the advantage of sensing larger area in the environment and hence capturing much detail all at one instant. Such imaging can be obtained through panoramic sensors which provide 360° field of view (FOV), or wide hemispherical FOV cameras such as the Eyefish camera (see figure 3.2). A classical method to obtain wide-imaging is by image stitching.

Image stitching is the process of combining multiple images with overlapping fields of view to produce a panorama (see figure B.1). A robot can generate this panoramic image by capturing several image shots through its sensor actuation. The overlapping image shots will be acquired while the camera is in motion. The camera can be actuated by parallel translational motion with respect to a wall for example, or can be rotated around its central axis to capture the surrounding view. The former represents an easy projection on a planar surface, while the latter represents projection on a cylindrical surface. Both techniques will not generate accurate stitching if the camera exhibits severe vibrations.

To perform the stitching, correspondences between the sequential image shots need to be found. While most of the older techniques work by directly minimizing pixel-to-pixel dissimilarities, a different class of algorithms works by extracting a sparse set of features and then matches them to each other. Most of the recent panorama creation methods and available commercial tools use the latter technique, in which the RANSAC [Fischler and Bolles, 1981] is employed to find the correspondences.

When directly matching pixel intensities is used, alignment between two images is established in order to shift one image relative to the other. The other approach, however, extracts distinctive features from each image, matches individual features to establish the global correspondence, and then estimates the geometric transformation between the images. This kind of approach has been used since the early days of stereo matching, and has recently gained more popularity for image stitching applications. Feature-based approaches have the advantage of being more robust against scene movement, and are potentially faster. Their biggest advantage is their ability to automatically discover the adjacency (overlap) relation-

Figure B.1: A set of images and the panorama discovered in them.

ships among an unordered set of images [Brown and Lowe, 2003]. Among the most successful employed features for generating panoramas is the Scale Invariant Feature Transform (SIFT) local detector and descriptor [Lowe, 2004].

The image stitching process can be divided into three main components: *image registration*, *calibration* and *blending*.

Image registration involves using direct alignment methods or matching features to search for the alignments that minimize the sum of absolute differences between overlapping pixels. Since there are a small number of common features between two sequential images, the result of the search is more accurate and matching execution is faster. *Image calibration* aims to minimize differences between an ideal lens models and the camera-lens that was used, to remove optical defects such as distortions. Once the features are registered and their absolute positions are recorded, this data can be used for the geometric optimization of the images. Panotools and its various derivative programs use this method. *Image blending* involves executing the adjustments figured out in the calibration stage, combined with images remapping to an output projection. Perspective wrapping is done through homographies, and finally, seam line adjustment is done to minimize the visibility of seams between stitched images.

The difficulty in the image stitching lies in the capture conditions. When the camera (accordingly the robot) undergoes rapid movement, severe vibrations or dynamic motion, artifacts will occur as a result of time differences between the image segments. "Blind stitching" through feature-based alignment methods, as opposed to manual selection and stitching, can cause imperfections in the assembly of the panorama.

Bibliography

[Andreasson and Duckett, 2004] H. Andreasson and T. Duckett, "Topological localization for mobile robots using omni-directional vision and local features," *in Proceedings IAV 2004, the 5th IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, 2004.

[Appin, 2007] Appin Knowledge Solutions, Robotics. Infinity Science Press LLC, 2007.

[Artac, 2002] M. Artac, "Mobile robot localisation with incremental PCA", *in Proceedings of the 11th Mediterranean Electrotechnical Conference (MELECON)*, pp. 192–197, 2002.

[Atiya and Hager, 1993] S. Atiya and G Hager, "Real-time vision-based robot localization." *in IEEE Transactions on Robotics and Automation*, 9(6): 785–800, 1993.

[Ayers and Boutell, 2007] B. Ayers and M. Boutell, "Home interior classification using SIFT keypoint histograms," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–6, 2007.

[Badreddin, 1987] E. Badreddin, "Recursive control structure for mobile robots," *International Conference on Intelligent Autonomous Systems (IAS)*, pp. 11–14, Amsterdam, December 1987.

[Badreddin, 1991] E. Badreddin, "Recursive behavior-based architecture for mobile robots," *Robotics and Autonomous Systems*, 8(3): 165–176, 1991.

[Badreddin, 1997] E. Badreddin, "Control and system design of wheeled mobile robots," Habilitation script, The Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 1997.

[Badreddin, 2002] E. Badreddin, "Information theoretic unified modelling of autonomous systems," *in proceedings of the 19th National Radio Science Conference (NRSC)*, pp. 41–50, 2002.

[Bailey, 2005] T. Bailey, "Mobile robot localisation and mapping in extensive outdoor environments," PhD Thesis. The University of Sydney, Australia, August 2002.

[Ballesta et al., 2007] M. Ballesta, A. Gil, O. M. Mozos, and O. Reinoso, "Local descriptors for visual SLAM," *in Workshop on Robotics and Mathematics*, Coimbra, Portugal, 2007.

[Baltzakis and Trahanias, 2002] H. Baltzakis and P. Trahanias, "Hybrid mobile robot localization using switching state-space models," *in Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 366–373, 2002.

[Bay et al., 2008] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, "SURF: Speeded Up Robust Features", *Computer Vision and Image Understanding (CVIU)*, 110(3): 346–359, 2008.

[Betke and Gurvits, 1997] M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *IEEE Transactions on Robotics and Automation*, 13(2): 251–263, 1997.

[Beeson et al., 2005] P. Beeson, N. K. Jong, and B. Kuipers, "Towards autonomous topological place detection using the extended voronoi graph," *in Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, 2005.

[Bennewitz et al., 2006] M. Bennewitz, C. Stachniss, W. Burgard, and S. Behnke, "Metric localization with scale-invariant visual features using a single perspective camera," *in European Robotics Symposium (EUROS), Ed., H. Christiensen, vol. 22 of STAR Springer tracts in advanced robotics*, Springer Verlag Berlin Heidelberg, Germany, 2006.

[Biber and Straßer, 2003] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," *in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, USA, pp. 2743 – 2748, October 2003.

[Biber et al., 2005] P. Biber, S. Fleck, F. Busch, M. Wand, T. Duckett, and W. Straßer, "3D modeling of indoor environments by a mobile platform with a laser scanner and panoramic camera," *in Proceedings of the 13th European Signal Processing Conference (EUSIPCO)*, 2005.

[Blaer and Allen, 2002] P. Blaer and P. K. Allen, "Topological mobile robot localization using fast vision techniques," *in Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1031–1036, 2002.

[Blanco, 2000] D. Blanco, B. L. Boada, L. Moreno, and M. A. Salichs, "Local map from on-line laser voronoi extraction," in Proceedings of the IEEE International Conference on Intelligence Robots and Systems, pp. 103–108, 2000.

[Blanco et al., 2006] J. L. Blanco, J. Gonzalez, and J. A. Fernández-Madrigal, "Consistent observation grouping for generating metric-topological maps that improves robot localization," *in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 818–823, 2006.

[Bonin-Font et al., 2008] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: a survey," *Journal of Intelligent and Robotic Systems*, 53(3): 263–296, 2008.

[Bonnifait and Garcia, 1996] P. Bonnifait and G. Garcia, "A multisensor localization algorithm for mobile robots and its real-time experimental validation,"" *in Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1395–1400, 1996.

[Borenstein et al., 1997] J. Borenstein, H. Everett, L. Feng, and D. Wehe, "Mobile robot positioning: Sensors and techniques," *Journal of Robotic Systems*, 14(4): 231–249, 1997.

[Borges and Aldon, 2004] G. A. Borges and M.-J. Aldon, "Line extraction in 2D range images for mobile robotics," *Journal of Intelligent and Robotic Systems*, 40(3): 267–297, 2004.

[Booij et al., 2006] O. Booij, Z. Zivkovic, and B. Kröse, "Sparse appearance based modelling for robot localization," *in Proceedings of the International Conference on Intelligent Robots and Systems*, pp. 1510–1515, 2006.

[Bossler et al., 2002] J. D. Bossler, J. R. Jensen, R. B. McMaster, and C. Rizos, Manual of Geospatial Science and Technology, CRC Press, London and New York, 2002, pp. 34–38.

[Briechle and Hanebeck, 2004] K. Briechle and U.D. Hanebeck, "Localization of a mobile robot using relative bearing measurements," *IEEE Transactions on Robotics and Automation*, 20 (1): 36–44, 2004.

[Briggs et al., 2000] Amy J. Briggs, D. Scharstein, D. Braziunas, C. Dima and P. Wall, "Mobile robot navigation using self-similar landmarks", *in Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1428 - 1434, 2000.

[Brown and Lowe, 2003] M. Brown and D. Lowe, "Recognizing panoramas," *in Proceedings of the 9th IEEE International Conference on Computer Vision*, vol. 2, pp. 1218–1227, 2003.

[Budenske, 1989] J Budenske, "Determining robot actions for tasks requiring sensor interaction", *in NASA Conference on Space Telerobotics*, 1989.

[Buschka and Saffiotti, 2004] P. Buschka and A. Saffiotti, "Some notes on the use of hybrid maps for mobile robots," *in the 8th International Conference on Intelligent Autonomous Systems (IAS)*, Amsterdam, pp. 547–556, 2004.

[Buschka, 2005] P. Buschka, "An investigation of hybrid maps for mobile robots," PhD Thesis. Örebro University, Örebro, Sweden, December 2005.

[Burgard et al., 1998] W. Burgard, A. Derr, D. Fox, and A.B. Cremers, "Integrating global position estimation and position tracking for mobile robots: The dynamic markov localization approach," *in IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 730–735, 1998.

[Calabrese and Indiveri, 2005] F. Calabrese and G. Indiveri, "An omni-vision triangulation-like approach to mobile robot localization," *in Proceedings of the IEEE International Symposium on Intelligent Control*, pp. 604–609, Cyprus, June 2005.

[Chakravarty, 2005] P. Chakravarty, "Vision-based indoor localization of a motorized wheelchair," *Technical Report MECSE-25-2005*, Monash University, 2005.

[Chandola, 2007] V. Chandola, A. Banerjee, and V. Kumar, "Outlier detection-a survey", *Tech. Report TR 07-017*, University of Minnesota, August 2007.

[Chatila and Laumond, 1985] R. Chatila and J. -P. Laumond, "Position referencing and consistent world modeling for mobile robots" *in Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 138–145, St. Louis, March 1985.

[Chen and Xu, 2006] H. Chen and Z. Xu, "3D map building based on stereo vision," *in Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC '06)*, pp. 969 – 973, August 2006.

[Choi et al., 2002] C.-H. Choi, J.-B. Song, W. Chung, and M. Kim, "Topological map building based on thinning and its application to localization," *in Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 552–557, 2002.

[Chong and Kleeman, 1999] K. Chong and L. Kleeman, "Mobile robot map building from an advanced sonar array and accurate odometry," *International Journal of Robotics Research*, 18(1): 20–36, 1999.

[Choset and Nagatani, 2001] H. Choset, and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization," *IEEE Transactions on Robotics and Automation*, 17(2): 125–137, 2001.

[Cohen and Koss, 1992] C. Cohen and F. V. Koss, "A comprehensive study of three object triangulation," *in Proceedings of the SPIE Conference on Mobile Robots*, pp. 95–106, 1992.

[Cole and Newman, 2006] D. M. Colea and P. Newman, "Using laser range data for 3D SLAM in outdoor environments," *in Proceedings IEEE International Conference on Robotics and Automation*, pp. 1556–1563, 2006.

[Cover and Thomas, 1991] T. M. Cover and J. A. Thomas. Elements of Information Theory. Wiley Series in Telecommunications, 1991.

[Crowley, 1989] J. L. Crowley, "World modeling and position estimation for a mobile robot using ultrasonic ranging," *in Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 674–680, May 1989.

[Courbon et al., 2008] J. Courbon, Y. Mezouar, L. Eck, and P. Martinet, "Efficient hierarchical localization method in an omnidirectional images memory," *in Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13–18, 2008.

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

[Davison, 2005] A. J. Davison, "Active search for real-time vision," *in Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV)*, pp. 66–73, 2005.

[Dellaert et al., 1999] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots," *in proceedings of the International Conference on Robotics and Automation (ICRA)*, 1999.

[Denzler and Brown, 2002] J. Denzler and C.M. Brown, "Information theoretic sensor data selection for active object recognition and state estimation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):145–157, 2002.

[Desouza and Kak, 2002] G.N Desouza and A.C Kak, "Vision for mobile robot navigation: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2): 237–267, 2002.

[Diosi and Kleeman, 2005] A. Diosi and L. Kleeman, "Laser scan matching in polar coordinates with application to SLAM," *in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3317–3322, August 2005.

[Duda et al., 2001] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification. John Wiley & Sons, New York, 2001, pp. 9–13.

[Duckett et al., 2002] T. Duckett, S. Marsland, and J. Shapiro, "Fast on-line learning of globally consistent maps," Autonomous Robots, 12(3):287–300, 2002.

[Duckett and Saffiotti, 2000] T. Duckett and A. Saffiotti, "Building globally consistent gridmaps from topologies," *in Proceedings of the 6th International IFAC Symposium on Robot Control*, Elsevier, pp. 357–361 , 2000.

[Durrant-Whyte and Bailey, 2006] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I the essential algorithms," *IEEE Robotics and Automation Magazine*, 13(2): 99–108, 2006.

[Easton and Cameron, 2006] A. Easton and S. Cameron, "A Gaussian error model for triangulation-based pose estimation using noisy landmarks," *in IEEE Conference on Robotics, Automation and Mechatronics*, pp. 1–6, June 2006.

[Elfes, 1989] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *IEEE Computer*, 22(6): 46–57, 1989.

[Elfes, 1990] A. Elfes, "Occupancy grids: A stochastic spatial representation for active robot perception," *in proceedings of the 6th Conference on Uncertainty in Artificial Intelligence*, pp. 136–146, 1990.

[Ellekilde et al., 2007] L.-P. Ellekilde, S. Huang, J. V. Miro, and G. Dissanayake, "Dense 3D map construction for indoor search and rescue," *Journal of Field Robotics*, 24(1-2): 71–89, 2007.

[Engelson and McDermott, 1992] S. Engelson and D. McDermott, "Error correction in mobile robot map learning," *in proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2555–2560, 1992.

[Esteves et al., 2003] J. S. Esteves, A. Carvalho and C. Couto, "Generalized geometric triangulation algorithm for mobile robot absolute self-localization," *in IEEE International Symposium on Industrial Electronics*, pp. 346–351, Brazil, June 2003.

[Esteves et al., 2006] J. S. Esteves, A. Carvalho and C. Couto, "Position and orientation errors in mobile robot absolute self-localization using an improved version of the generalized geometric triangulation algorithm," *in IEEE International Conference on Industrial Technology, (ICIT)*, pp. 830–835, 2006.

[Fabrizi and Saffiotti, 2000] E. Fabrizi and A. Saffiotti, "Extracting topology-based maps from gridmaps," *in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2972–2978, San Francisco, CA, 2000.

[Filliat and Meyer, 2003] D. Filliat and J.-A. Meyer, "Map-based navigation in mobile robots: I. A review of localization strategies," *Journal of Cognitive System Research*, 4: 283-317, 2003.

[Filliat, 2007] D. Filliat, "A visual bag of words method for interactive qualitative localization and mapping," *in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3921–3926, 2007.

[Fischler and Bolles, 1981] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography," Communication Association and Computing Machine, 24(6): 381–395, 1981.

[Font-Llagunes and Batlle, 2009] J. M. Font-Llagunes, J. A. Batlle, "Consistent triangulation for mobile robot localization using discontinuous angular measurements," *Robotics and Autonomous Systems*, 57(9): 931–942, 2009.

[Font and Batlle, 2006] J.M. Font and J.A. Batlle, "Mobile robot localization: revisiting the triangulation methods," *in Proceedings of the IFAC Symposium on Robot Control*, 2006.

[Fox et al., 1998] D. Fox, W. Burgard, and S. Thrun, "Active markov localization for mobile robots," *in Robotics and Autonomous Systems*, 25: 195–207, 1998.

[Fox et al., 1999] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *Journal of Artificial Intelligence Research*, 11: 391–427, 1999.

[Galindo et al., 2005] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.A. Fernández-Madrigal, and J. González, "Multi-hierarchical semantic maps for mobile robotics," *in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3492–3497, 2005.

[Giesbrecht, 2004] J. Giesbrecht, "Global path planning for unmanned ground vehicles," Technical Memorandum TM 2004-272, Defence R&D Canada, Suffield, 2004.

[Goncalves et al., 2005] L. Goncalves, E. Di Bernardo, D. Benson, M. Svedman, J. Ostrowski, N. Karlsson, and P. Pirjanian, "A visual frontend for simultaneous localization and mapping," *in Proceedings of the International Conference on Robotics and Automation*, pp. 44–49, 2005.

[Grisetti et al., 2007] G. Grisetti, G. D. Tipaldi, C. Stachniss, W. Burgard, and D. Nardi, "Fast and accurate SLAM with Rao-blackwellized particle filters," *in Journal of Robotics and Autonomous Systems*, 55: 30 –38, 2007.

[Gross et al., 2009] H.-M. Gross, H.-J. Böhme, Ch. Schröter, St. Müller, A. König, E. Einhorn, Ch. Martin, M. Merten, and A. Bley, "TOOMAS: interactive shopping guide robots in everyday use - final implementation and experiences from long-term field trials," *in Proceedings of the IEEE/RJS International Conference on Intelligent Robots and Systems*, pp. 2005-2012, 2009.

[Guivant et al., 2004] J. Guivant, E. Nebot, J. Nieto, and F. Masson, "Navigation and mapping in large unstructured environments," *in International Journal of Robotics Research*, 23:449–472, April 2004.

[Guyon and Elisseeff, 2003] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research. Special Issue on Variable and Feature Selection*, 3:1157–1182, 2003.

[Hamner et al., 2009] B. Hamner, S. Koterba, J. Shi, R. Simmons, and S. Singh, "Mobile robotic dynamic tracking for assembly tasks source**,"** *in proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, USA, pp. 2489–2495, 2009.

[Han and Kamber, 2006] Jiawei Han and Micheline Kamber. Data Mining: Concepts and Techniques, 2nd Edition, Morgan Kauffman, 2006.

[Harris and Stephens, 1988] C. Harris and M. Stephens, "A combined corner and edge detector," *in Proceedings of the 4th Alvey Vision Conference*, pp. 147–151, 1988.

[Hernandez et al., 2003] S. Hernandez, C. Morales, J. Torres, and L. Acosta, "A new localization system for autonomous robots," *in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taiwan, pp. 1588–1593, 2003.

[Ho and Newman, 2005a] K. Ho, P. Newman, "Combining visual and spatial appearance for loop closure detection in SLAM," *in proceedings of the 2nd European Conference on Mobile Robots (ECMR)*, Italy, September 2005.

[Ho and Newman, 2005b] K. Ho and P. Newman, "Multiple map intersection detection using visual appearance**,"** *in proceedings of the 3rd International Conference on Computational Intelligence, Robotics and Autonomous Systems*, Singapore, December 2005

[Ho and Newman, 2007] K. L. Ho and P. Newman, "Detecting loop closure with scene sequences," *International Journal of Computer Vision*, 74(3): 261–286, 2007.

[Jang et al., 2003] G. Jang, S. Kim, W. Lee and I. Kweon, "Robust self-localization of mobile robots using artificial and natural landmarks," *in Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 412–417, 2003.

[Jang et al., 2005] G. Jang, S. Kim, J. Kim, and I. Kweon, "Metric localization using a single artificial landmark for indoor mobile robots," *in IEEE International Conference on Intelligent Robots and Systems*, pp. 2857–2862, 2005.

[Jebara et al., 1999] T. Jebara, A. Azarbayejani, and A. Pentland, "3D structure from 2D motion," *IEEE Signal Processing Magazine*, 16(3): 66 - 84, 1999.

[Jensfelt and Kristensen, 2001] P. Jensfelt and S. Kristensen, "Active global localisation for a mobile robot using multiple hypothesis tracking," *IEEE transactions on Robotics and Automation*, 17(5): 748–-760, October 2001.

[Jipp et al., 2005] M. Jipp, E. Badreddin, C. Bartolein, A. Wagner, and W. W. Wittmann, "Cognitive modeling to enhance usability of complex technical systems in rehabilitation technology on the example of a wheelchair," *in proceedings of the 2nd International Conference on Mechatronics*, vol. 2, pp. 1165–1172, 2005.

[Jung and Kim, 2005] D. J. Jung and H. J. Kim, "Place recognition system from long-term observations," *in Proceedings of the 18th International Conference on Innovations in Applied Artificial Intelligence*, pp. 36–43, 2005.

[Jurie and Triggs, 2005] F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," *in Proceedings of the 10th IEEE International Conference on Computer Vision*, pp. 604–610, 2005.

[Ke and Sukthankar, 2004] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," *in Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 506–513, 2004.

[Kelly, 2003] A. Kelly, "Precision dilution in triangulation based mobile robot position estimation," *in Proceedings of the International Conference on Intelligent Autonomous Systems*, Amsterdam, 2003.

[Koenig and Simmons, 1996] S. Koenig and R. Simmons, "Unsupervised learning of probabilistic models for robot navigation," *in proceedings of the IEEE Conference on Robotics and Automation (ICRA)*, pp. 2301–2308, 1996.

[Kosecka and Li, 2004] J. Kosecka and F. Li, "Vision based topological markov localization," *in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1481–1486, 2004.

[Krotkov, 1989] E. Krotkov, "Mobile robot localization using a single image," *in Proceedings of the International Conference on Robotics and Automation*, pp. 978–983, 1989.

[Kröse et al., 2001] B. Kröse, O. Vlassis, R. Bunschoten, and Y. Motomura, "A probabilistic model for appearance-based robot localization," *Image and Vision Computing*, 19(6):381–391, 2001.

[Kuipers, LN18-19] B. Kuipers, "Robotics," *Lecture notes*, CS 344R/393R, University of Texas, Lectures 18 and 19.

[Kuipers, 1978] B. J. Kuipers, "Modeling spatial knowledge," *Cognitive Science*, 2(2): 129–153, 1978.

[Kuipers and Beeson, 2002] B. Kuipers and P. Beeson, "Bootstrap learning for place recognition," *in AAAI Conference on Artificial Intelligence*, pp. 174–180, 2002.

[Kuipers, 2008] B. Kuipers, "An intellectual history of spatial semantic hierarchy," *in Robot and Cognitive Approaches to Spatial Mapping*, Springer Tracts in Advanced Robotics, Jefferies, M. and Yeap, A. W.-K.(eds), Berlin Springer, vol. 38, pp. 243–264, 2008.

[Kwok et al., 2003] C. T. Kwok, D. Fox, and M. Meila, "Adaptive real-time particle filters for robot localization," *in Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2836–2841, 2003.

[Kwon et al., 2008] B. Kwon, T. Choi, H. Chung, and G. Kim, "Parallelization of the scale-invariant keypoint detection algorithm for cell broadband engine architecture," *in Proceedings of the 5th IEEE Consumer Communications and Networking Conf. (CCNC 08)*, pp. 1030–1034, 2002.

[Laaksonen, 2007] J. Laaksonen, "Mobile robot localization using sonar ranging and WLAN intensity maps," Bachelor's thesis, Lappeenranta University of Technology, February 2007.

[Lamon et al. 2003] P. Lamon, A. Tapus, E. Glauser, N. Tomatis, and R. Siegwart, "Environmental modeling with fingerprint sequences for topological global localization," *in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 4, pp. 3781–3786, 2003.

[Lungarella and Pfeifer, 2001] M. Lungarella and R. Pfeifer, "Robots as cognitive tools: information theoretic analysis of sensory-motor data," in Proceedings of the IEEE-RAS International Conference on Humanoid Robots, pp. 245-252, 2001.

[Lankenau and Rofer, 2002] A. Lankenau and T. Rofer, "Mobile robot self-localization in large-scale environments," *in proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2. Washington D.C, pp. 1359–1364, May 2002.

[Ledwich and Williams, 2004] L. Ledwich and S. Williams, "Reduced SIFT features for image retrieval and indoor localisation," *in Proceedings of Australasian Conference on Robotics and Automation (ACRA)*, 2004.

[Lee and Song, 2007] S. Lee and J-B. Song, "Use of coded infrared light as artificial landmarks for mobile robot localization," *in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1731–1736, 2007.

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

[Lewis, 1995] D. D. Lewis, "Evaluating and optimizing autonomous text classification systems," *in Proceedings of the 18th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 246–254, 1995.

[Li and Kosecka, 2006] F. Li and J. Kosecka, "Probabilistic location recognition using reduced feature set," *in Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3405–3410, 2006.

[Lindeberg, 1998] T. Lindeberg, "Feature detection with automatic scale selection'," *International Journal of Computer Vision*, 30(2): 77–116, 1998.

[Lingemann et al., 2005] K. Lingemann, H. Surmann, A. Nuchter, and J. Hertzberg, "High-speed laser localization for mobile robots," *Journal of Robotics and Autonomous Systems*, 51(4): 275–296, 2005.

[Lisien et al., 2003] B. Lisien, D. Morales, D. Silver, G. Kantor, I. Rekleitis, and H. Choset, "Hierarchical simultaneous localization and mapping," in IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 448–453, 2003.

[Loncomilla and Ruiz-del-Solar, 2005] P. Loncomilla and J. Ruiz-del-Solar, "Improving SIFT-based object recognition for robot applications," *in 13th International Conference on Image Analysis and Processing (ICIAP)*, pp. 1084–1092, 2005.

[Lowe, 1999] D. Lowe, "Object recognition from local scale-invariant features," *in Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV)*, vol. 2, pp. 1150–1157, 1999.

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

[Lu and Milios, 1997] F. Lu and E. Milios, "Robot pose estimation in unknown environments by matching 2D range scans," *Journal of Intelligent and Robotic System*, 18(3): 249–275, 1997.

[Luo et al., 2006] J. Luo, A. Pronobis, B. Caputo, and P. Jensfelt, The KTH-IDOL2 database, Technical Report CVAP304, Kungliga Tekniska Hoegskolan, CVAP/CAS, October 2006.

[Luo et al., 2007] J. Luo, A. Pronobis, B. Caputo, and P. Jensfelt, "Incremental learning for place recognition in dynamic environments," *in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 721–728, 2007.

[Madsen et al., 1997] C. B. Madsen, C. S. Andersen and J. S. Sørensen, "A robustness analysis of triangulation-based robot self positioning," *in Proceedings of the International Symposium on Intelligent Robotic Systems*, Sweden, 1997.

[Madsen and Andersen, 1998] C. Madsen and C. Andersen, "Optimal landmark selection for triangulation of robot position," *Journal of Robotics and Autonomous Systems*, 13(4), pp. 277–292, 1998.

[Meng et al, 2000] Q.-H. Meng, Y.-C. Sun, and Z.-L. Cao, "Adaptive extended Kalman filter (AEKF)-based mobile robot localization using sonar," *Robotica*, 18(5): 459–473, 2000.

[Menegatti et al., 2003a] E. Menegatti, M. Zoccarato, E. Pagello, and H. Ishiguro, "Image-based Monte-Carlo localisation without a map," *in Proceedings of the 8th Conference of the Italian Association for Artificial Intelligence*, pp. 423–435, 2003.

[Menegatti et al., 2003b] E. Menegatti, M. Zoccarato, E. Pagello, and H. Ishiguro, "Hierarchical image-based localisation for mobile robots with Monte-Carlo localisation," *in Proceedings of the European Conference on Mobile Robots*, pp. 13–20, 2003.

[Menegatti et al., 2004] E. Menegatti, T. Maeda, and H. Ishiguro, "Image-based memory for robot navigation using properties of the omnidirectional images," *in Robotics and Autonomous Systems*, 47(4): 251–267, 2004.

[Menegatti et al., 2006] E. Menegatti, A. Pretto, A. Scarpa, and E. Pagello, "Omnidirectional vision scan matching for robot localization in dynamic environments," *IEEE Transactions on Robotics*, 22:523–535, 2006.

[Messom and Barczak, 2008] C. Messom and C. Barczak, "Stream processing of integral images for real-time object detection," *in proceedings of 9th International Conference on Parallel and Distributed Computing, Applications and Technologies*, pp. 405–412, 2008.

[Mikolajczyk and Schmid, 2003] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *in Proceedings of IEEE Confernce on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 257–263, 2003.

[Mikolajczyk and Schmid, 2004] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, 60(1): 63–86, 2004.

[Min et al., 2007] S. Min, H. Rixin, and W. Daojun, "Precision analysis to 3D reconstruction from image sequences," *in ISPRS Workshop on Updating Geo-spatial Databases with Imagery & The 5th ISPRS Workshop on DMGISs*, pp.141–146, 2007.

[Moravec and Elfes, 1985] H. P. Moravec and A. Elfes, "High resolution maps from wide angle sonar," *in Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 116–121, 1985.

[Mozos et al., 2005] O. M. Mozos, C. Stachniss, and W. Burgard, "Supervised learning of places from range data using AdaBoost," *in Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1742–1747, 2005.

[Mozos et al. 2006] O. M. Mozos, A. Rottmann, R. Triebel, P. Jensfelt, and W. Burgard, "Semantic labeling of places using information extracted from laser and vision sensor data," *in IEEE/RSJ Intelligent Robots and Systems Workshop: From sensors to human spatial concepts*, 2006.

[Mozos et al., 2007] O. M. Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard, "Supervised semantic labeling of places using information extracted from sensor data," *Robotics and Autonomous Systems*, 55(5): 391–402, 2007.

[Murillo et al., 2007a] A. C. Murillo, J. J. Guerrero, and C. Sagues, "Surf features for efficient robot localization with omnidirectional images," *in Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3901–3907, 2007.

[Murillo et al., 2007b] A. C. Murillo, C. Sagüés, J. J. Guerrero, T. Goedemé, T. Tuytelaars, and L. V. Gool, "From omnidirectional images to hierarchical localization," *in Robotics and Autonomous Systems*, 55(5): 372–382, 2007.

[Nebot and Pagac, 1995] E. M. Nebot and D. Pagac, "Quadtree representation and ultrasonic information for mapping an autonomous guided vehicle's environment," *in International Journal of Computers and Their Applications*, 2(3):160–170, 1995.

[Nüchter et al., 2006] A. Nüchter, K. Lingemann, and J. Hertzberg, "Extracting Drivable Surfaces in Outdoor 6D SLAM," *in Proceedings of the 37th International Symposium on Robotics (ISR '06) and the 4th German Conference Robotik 2006*, ISBN 3-18-091956-6, Munich, Germany, 2006.

[Ohya et al., 1994] A. Ohya, Y. Nagashima, and S.-I. Yuta, "Exploring unknown environment and map construction using ultrasonic sensing of normal direction of walls," *in IEEE International Conference on Robotics and Automation*, pp. 485–492, 1994.

[Oliva and Torralba, 2001] A. Oliva and A .Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *in International Journal of Computer Vision*, 42(3): 145–175, 2001.

[Olsson et al., 2004] L. Olsson, C.L. Nehaniv, and D. Polani, "The effects on visual information in a robot in environments with oriented contours," *in L. Berthouze, H. Kozima, C.G. Prince, G. Sandini, G. Stojanov, and G. Metta, editors, Proceedings of the 4th International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, pp. 83–88, Genoa, Italy, 2004.

[Pèrez, 2005] D. H. Pèrez, H. M. Barberà, and A. Saffiotti, "Fuzzy self-localization using natural features in the four-legged league," in Robot Soccer World Cup, ser. LNCS, pp. 110–121, 2005.

[Pfister et al., 2003] S. Pfister, S. Roumeliotis, and J. Burdick, "Weighted line fitting algorithms for mobile robot map building and efficient data representation," *in Proceedings of the IEEE International Conference on Robotics and Automation*, Taiwan, pp. 1304–1311, September 2003.

[Piasecki, 1995] M. Piasecki, "Global localization for mobile robots by multiple hypothesis tracking", *Robotics and Autonomous Systems*, 16, pp. 93–104, 1995.

[Pierlot and Droogenbroeck, 2009] V. Pierlot and M. V. Droogenbroeck, "Simple and low cost angle measurement system for mobile robot positioning," *in 20th Annual Workshop on Circuits, Systems and Signal Processing (ProRISC)*, pp. 251–254, 2009.

[Polani et al., 2001] D. Polani, T. Martinetz, and J. Kim, "An information-theoretic approach for the quantification of relevance," *in Proceedings of the 6th European Conference on Advances in Artificial Life*, Springer LNCS, 2159, pp. 704–713, 2001.

[Prassler, 2001] E. Prassler, J. Scholz and P. Fiorini, "A robotic wheelchair for crowded public environments," IEEE Robotics and Automation Magazine, 8(1):38–45, 2001.

[Press, 1988] W. Press, W. Vetterling, S. Teukolosky, and B. Flannery. Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, second edition, 1988.

[Pronobis et al., 2006] A. Pronobis, B. Caputo, P. Jensfelt, and H.I. Christensen, "A discriminative approach to robust visual place recognition," *in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3829–3836, 2006.

[Pronobis and Caputo, 2009] A. Pronobis and B. Caputo, "COLD: COsy Localization Database," *International Journal of Robotics Research (IJRR)*, 28(5):588–594, May 2009.

[Pronobis et al., 2010] A. Pronobis, B. Caputo, P. Jensfelt, H.I. Christensen, "A realistic benchmark for visual indoor place recognition," *Robotics and Autonomous Systems*, 58(1): 81–96, January 2010.

[Pronobis, 2011] A. Pronobis, "Semantic mapping with mobile robots," PhD thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, June 2011.

[Qian et al., 2004] G. Qian, S. Sural, Y. Gu, and S. Pramanik, "Similarity between Euclidean and cosine angle distance for nearest neighbor queries," *in proceedings of the ACM symposium on Applied computing*, pp. 1232–1237, 2004.

[Radhakrishnan and Nourbakhsh, 1999] D. Radhakrishnan and I. Nourbakhsh, "Topological robot localization by training a vision-based transition detector," *in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 468–473, 1999.

[Rady et al., 2008] S. Rady, A. Wagner, and E. Badreddin, "Entropy-based Features for Robust Place Recognition," *in Proceedings of the IEEE International Conference on System, Man and Cybernetics (SMC08)*, Singapore, October 2008, pp. 713-718.

[Rady et al., 2009] S. Rady, A. Wagner, and E. Badreddin, "Efficient Codebook Generation for Appearance-based Localization," *in Proceedings of the Asian Control Conference (ASCC09)*, Hong Kong, August 2009, pp. 1656-1661.

[Rady et al., 2010a] S. Rady, A. Wagner, and E. Badreddin, "Building Efficient Topological Maps for Mobile Robot Localization: An Evaluation Study on COLD Benchmarking Database," *in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS10)*, Taiwan, October 2010, pp. 542-547.

[Rady et al., 2010b] S. Rady, A. Wagner, and E. Badreddin, "Hierarchical Localization using Entropy-based Feature Maps and Triangulation Techniques," *in Proceedings of the IEEE International Conference on System, Man and Cybernetics (SMC10)*, Istanbul, Turkey, October 2010, pp. 519-525.

[Rady and Badreddin, 2010] S. Rady and E. Badreddin, "Information-Theoretic Environment Modeling for Efficient Topological Localization," *in Proceedings of the 10th International Conference on Intelligent Systems Design and Applications (ISDA10)*, Cairo, Egypt, November 2010, pp. 1042-1046.

[Rady et al., 2011] S. Rady, A. Wagner, and E. Badreddin, "Hierarchical Localization using Compact Hybrid Mapping for Large-Scale Unstructured Environments," *in the IEEE International Conference on System, Man and Cybernetics (SMC11)*, Anchorage, Alaska, October 2011.

[Ramisa et al., 2008] A. Ramisa, A. Tapus, R. Lopez de Mantaras, R. Toledo, "Mobile robot localization using panoramic vision and combinations of feature region detectors," *in IEEE International Conference on Robotics and Automation*, pp. 538–543, 2008.

[Ramisa, 2009] A. Ramisa, "Localization and object recognition for mobile robots," PhD thesis, Universitat Autònoma de Barcelona, Bellaterra, July 2009.

[Remolina and Kuipers, 2004] E. Remolina and B. Kuipers, "Towards a general theory of topological maps," *Artificial Intelligence*, 152(1): 47–104, January 2004.

[Rocha et al., 2005] R. Rocha, J. Dias, and A. Carvalho, "Cooperative multi-robots systems: A study of vision-based 3D mapping using information theory," *in proceedings of the International Conference on Robotics and Automation*, pp. 384–389, 2005.

[Roth and Winter, 2008] P. M. Roth and M. Winter, "Survey of Appearance-based methods for object recognition," Technical Report, ICG-TR-01/08, Graz University of Technology, Institute for Computer Graphics and Vision.

[Rottmann et al., 2005] A. Rottmann, O. M. Mozos, C. Stachniss, and W. Burgard, "Semantic place classification of indoor environments with mobile robots using boosting," *in AAAI Conference on Artificial Intelligence*, pp. 1306–1311, 2005.

[Sabatta, 2008] D. G. Sabatta, "Vision-based topological map building and localisation using persistent features," *in Robotics and Mechatronics Symposium (ROBMECH)*, Pretoria, South Africa, pp. 1–6, 2008.

[Sandberg et al., 2009] D. Sandberg, K. Wolff, and M. Wahde, "A robot localization method based on laser scan matching," Lecture notes in computer science - advances in robotics, vol. 5744, Springer Berlin Heidelberg , pp. 171–178, 2009.

[Sala et al., 2004] P. Sala, R. Sim, A. Shokoufandeh, and S. Dickinson, "Landmark selection for vision-based navigation," *in Proceedings of the International Conference on Intelligent Robots and Systems*, pp. 3131–3138, 2004.

[Schiele and Crowley, 1998] B. Schiele and J.L. Crowley, "Transinformation for active object recognition," *in International Conference on Computer Vision (ICCV)*, pp. 249–254, Bombay, India, 1998.

[Schiele and Pentland, 1999] B. Schiele and A. Pentland, "Probabilistic object recognition and localization," *in International Conference on Computer Vision (ICCV)*, pp. 177–182, 1999.

[Schultz and Adams, 1998] A.C. Schultz and W. Adams, "Continuous localization using evidence grids," *in IEEE International Conference on Robotics and Automatio*n, pp. 2833–2839, 1998.

[Se et al., 2001] S. Se, D. G. Lowe, and J. J. Little, "Vision-based mobile robot localization and mapping using scale-invariant features," *in Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2051–2058, 2001.

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

[Seekircher et al., 2011] A. Seekircher, T. Laue, and T. Röfer, "Entropy-based active vision for a humanoid soccer robot," in J. Ruiz-del-Solar, E. Chown and P. G. Ploeger (Eds.): RoboCup 2010, Lecture Notes in Artificial Intelligence, Springer, pp. 1–12, 2011.

[Shannon, 1948] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, 27: 379–423, and 28: 623–656, July, October, 1948.

[Shatkay and Kaelbling, 1997] H. Shatkay and L. Kaelbling, "Learning topological maps with weak local odometric information," *in proceeding of the International Joint Conference on Artificial Intelligence (IJCAI97)*, pp. 920–929, 1997.

[Shatkay and Kaelbling, 2002] H. Shatkay and L. P. Kaelbling, "Learning geometrically-constrained hidden markov models for robot navigation: bridging the topological-geometric gap," *Journal of Artificial Intelligence Research*, 16: 167–207, 2002.

[Shoval and Sinriech, 2001] S. Shoval and D. Sinriech, "Analysis of landmark configuration for absolute positioning of autonomous vehicles," *Journal of Manufacturing Systems*, 20(1): 44–54, 2001.

[Siadat and Vialle, 2002] A. Siadat and S. Vialle, "Robot localization using P-similar landmarks, optimized triangulation and parallel programming," *in 2nd IEEE International Symposium on Signal Processing and Information Technology*, Morocco, December 2002.

[Siegwart and Nourbakhsh, 2004] R. Siegwart and I. R. Nourbakhsh. Introduction to Autonomous Mobile Robots. MIT Press Cambridge, Massachusetts, London, England, 2004, chapter 5.

[Siciliano and Khatib, 2008] B. Siciliano and O. Khatib. Springer Handbook of Robotics, (Eds.), Part E: Mobile and Distributed Robotics, World Modeling, Springer-Verlag Berlin Heidelberg, 2008.

[Sim, 1998] R. Sim, "Mobile robot localization from learned landmarks," Master's Thesis, McGill University, July 1998.

[Singhal, 1997] A. Singhal, "Issues in autonomous mobile robot navigation," Report of Computer Science Department, University of Rochester, May 1997.

[Smith et al., 1990] R. Smith, M. Self and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," *Autonomous Robot Vehicles*, I.J. Cox and G.T. Wilfon, editors, Springer-Verlag, pp. 167–193, 1990.

[Sujan and Dubowsky, 2005] V.A. Sujan and S. Dubowsky, "Visually guided cooperative robot actions based on information quality," *in Autonomous Robots*, Springer, 19(1): 89–110, 2005.

[Surmann et al., 2003] H. Surmann, A. Nuchter, 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*, 45(3-4): 181–198, December 2003.

[Tapus and Siegwart, 2005] A. Tapus and R. Siegwart, "Incremental robot mapping with fingerprints of places," *in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2429–2434, 2005.

[Thrun and Bücken, 1996] S. Thrun and A. Bücken, "Integrating grid-based and topological maps for mobile robot navigation", *in Proceedings of the 13th National Conference on Artificial Intelligence (AAAI)*, pp. 944–950, 1996.

[Thrun, 1998] S. Thrun, "Learning maps for indoor mobile robot navigation," Artificial Intelligence, vol. 99, pp. 21– 1, 1998.

[Thrun et al., 1998] S. Thrun, S. Gutmann, D. Fox, W. Burgard, and B. J. Kuipers, "Integrating topological and metric maps for mobile robot navigation: A statistical approach," *in Proceedings of the 15th National Conference on Artificial Intelligence*, pp. 989–995, 1998.

[Thrun et al., 2001] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artificial Intelligence Journal*, 128(1-2): 99–141, 2001.

[Thrun, 2002] S. Thrun, "Robotic mapping: A survey," in G. Lakemeyer and B. Nebel, editors, Exploring Artificial Intelligence in the New Millenium. Morgan Kaufmann, 2002.

[Thrun, 2003] S. Thrun, "Learning occupancy grids with forward sensor models," *Autonomous Robots*, 15:111–127, 2003.

[Thrun et al., 2005] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics. MIT Press, Cambridge, MA, 2005.

[Tomatis et al., 2003] N. Tomatis, I. Nourbakhsh, and R. Siegwart, "Hybrid simultaneous localization and map building: a natural integration of topological and metric," *in Robotics and Autonomous Systems*, 44(1): 3–14, 2003.

[Tomono, 2004] M. Tomono, "A scan matching method using Euclidean invariant signature for global localization and map building," *in Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 886–871, 2004.

[Torr, 2002] P.H.S. Torr, "A Structure and Motion Toolkit in Matlab," *Microsoft Research Technical Report No. MSR-TR-2002-56*, Redmond, WA, May 2002.

[Tully et al., 2007] S. Tully, H. Moon, D. Morales, G. Kantor, and H Choset, "Hybrid localization using the hierarchical atlas," *in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, pp. 2857–2864, 2007.

[Ulrich and Nourbakhsh, 2000] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," *in Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1023–1029, 2000.

[Ullah et al., 2008] M. M. Ullah, A. Pronobis, B. Caputo, J. Luo, R. Jensfelt, and H. I. Christensen, "Towards robust place recognition for robot localization," *in proceedings of the IEEE International Conference in Robotics and Automation*, pp.530–537, 2008.

[URL:COLD] "The COLD (COsy Localization Database) database," [Online]. Available at: http://cogvis.nada.kth.se/COLD/ , May 2009 [July 16th , 2011].

[URL:SPECTRUM] http://spectrum.ieee.org/automaton/robotics/industrial-robots/041410-world-robot-population , April 14th, 2010 [July 16th, 2011].

[URL:IFR] http://www.ifr.org/service-robots/statistics/ , [July 16th, 2011].

[URL:MET] http://www.nikonmetrology.com/ , [July 16th, 2011].

[Valgren, 2007] C. Valgren, "Topological mapping and localization using omnidirectional vision," Licentiate thesis, Örebro University, 2007.

[Valgren and Lilienthal, 2007] C. Valgren and A. Lilienthal, "Sift, surf and seasons: Long-term outdoor localization using local features," *in Proceedings of 3rd European Conference on Mobile Robots (ECMR)*, Freiburg, Germany, pp. 1–6, 2007.

[Vasudevan et al., 2006] S. Vasudevan, V. Nguyen, and R. Siegwart, "Towards a cognitive probabilistic representation of space for mobile robots," *in Proceedings of the IEEE International Conference on Information Acquisition (ICIA)*, pp. 353–359, 2006.

[Vasudevan et al., 2007] S. Vasudevan, S. Gächter, A. Harati, and R. Siegwart, "A hierarchical concept oriented representation for spatial cognition in mobile robots," M. Lungarella et al. (Eds.): 50 Years of AI, Festschrift, LNAI 4850, 244–257, 2007.

[Verdu, 1998] S. Verdu, "Fifty years of shannon theory," *IEEE Transactions on Information Theory*, 44(6): 2057–2078, October 1998.

[Wallgrün, 2004] J. O. Wallgrün, "Hierarchical Voronoi-based route graph representations for planning, spatial reasoning, and communication," *in proceedings of the 4th International Cognitive Robotics Workshop (CogRob)*, pp. 64–69, 2004.

[Wang et al., 2006] J. Wang, H. Zha and R. Cipolla, "Coarse-to-fine vision-based localization by indexing scale-Invariant features," *IEEE Transactions of Systems, Man, and Cybernetics*, pp. 413–422, 2006.

[Warren, 2007] L. Warren, "On Modelling Hybrid Uncertainty in Information," DSTO-RR-0325 Report, Defence Science and Technology Organisation, Australia, 2007.

[Welch and Bishop, 2006] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," University of North Carolina at Chapel Hill, Department of Computer Science, TR 95-041, July 2006.

[Welch et al., 1991] S. S. Welch, R. C. Montgomery, and M. F. Barsky, "The spacecraft control laboratory experiment optical attitude measurement system," *NASA technical memorandum* 102624, March 1991.

[Werner, 2010] F. Werner, "Vision-based topological mapping and localisation," PhD thesis, Queensland University of Technology, 2010.

[Wolf et al., 2005] J. Wolf, W. Burgard, and H. Burkhardt, "Robust vision-based localization by combining an image retrieval system with Monte Carlo localization," *in IEEE Transactions on Robotics*, pp. 208–216, 2005.

[Wu and Rehg, 2010] J. Wu and J. M. Rehg, "CENTRIST: a visual descriptor for scene categorization," *in IEEE Transactions PAMI*, pp. 2782–2783, 2010.

[Yairi et al., 2002] T. Yairi, M. Togami, K. Hori, "Learning Topological Maps from Sequential Observation and Action Data under Partially Observable Environment", *in the 7th Pacific Rim International Conference on Artificial Intelligence*, pp.305–314, 2002.

[Yang et al., 2004] J. Yang, D. Zhang, A. F. Frangi, and J. y. Yang, "Two-dimensional pca: A new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):131–137, 2004.

[Yguel et al., 2006] M. Yguel and O. Aycard and C. Laugier, "Efficient GPU-based construction of occupancy grids using several laser range-finders," *in IEEE International Conference on Intelligent Robot and Systems*, pp. 105–100, 2006.

[Yoon et al., 2007] K. Yoon, O. Kwon, and D. Bae, "An approach to outlier detection of software measurement data using the k-means clustering method," *in First International Symposium on Empirical Software Engineering and Measurement*, pp. 443–445, 2007.

[Yuen and MacDonald, 2005] D. C. K. Yuen, and B.A. MacDonald, "Vision-based localization algorithm based on landmark matching, triangulation, reconstruction, and comparison," *in IEEE Transactions on Robotics*, 21(2): 217–226, April 2005.

[Zhang et al., 2007] Z. Zhang, S. Chan, and L. T. Chia, "Codebook+: A new module for creating discriminative codebooks," *in proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 815–818, 2007.

[Zhang et al., 2008] Q. Zhang, Y. Chen, Y. Zhang, and Y. Xu; "SIFT implementation and optimization for multi-core systems,'' *in IEEE International Symposium on Parallel and Distributed Processing*, pp. 1–8, 2008.

[Zavlangas and Tzafestas, 2002] P. G. Zavlangas and S. G. Tzafestas, "Integration of topological and metric maps for indoor mobile robot path planning and navigation," *in proceedings of the 2nd Hellenic Conference on AI: Methods and Applications of Artificial Intelligence*, pp. 121–130, 2002.

[Zivkovic et al., 2005] Z. Zivkovic and B. Bakker and B. Kröse, "Hierarchical map building using visual landmarks and geometric constraints," *in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7–12, 2005.