Bio-inspired Navigation Based on Egocentric Visual Perception

Changmin Lee

The Graduate School
Yonsei University
School of Electrical and Electronic Engineering

Bio-inspired Navigation Based on Egocentric Visual Perception

A Dissertation Thesis
Submitted to the School of Electrical and Electronic Engineering
and the Graduate School of Yonsei University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Changmin Lee

February 2018

감사의 글

2018년 2월

2007년 추위속에 청운의 꿈으로 동서울 터미널에 내렸던 날을 기억합니다. 그날로부터 연세대학교에서 보낸 11년이라는 시간은 수많은 추억을 만들었습니다. 다양한 추억속에서 많은 사람을 만났고, 소중한 인연을 통해 인생에서 가장 풍부한 추억으로 20대의 일기장을 아름답게 채웠습니다. 그리고 수 많은 이야기들 중, 연구실에서의 6년 6개월간의 시간이 단연코 최고의 주인공입니다. 이는 제 지식에 있어, 갑자기 주어진 20살의 자유를 착각한 방종에 대해 책임을 지는 시간이었고, 자기 능력에 대한 자만을 회개하고 겹허히 새로운 지식을 채워나가는 과정이었습니다. 모든 반성과 수련이 그러하듯 노력의 시간에 동반되는 고통은 지금도 악몽을 꿀 정도로 힘들었지만, 배움의 책임에 대해 그만큼 분에 넘치는 기회였고 여유였으며 종극에 이르러 졸업을 앞에 두고선 '그래도 행복했노라' 자신있게 말할 수 있습니다. 물론 제가 가지 못한 길들과 다른 친우들의 길에 대해 건방지게 옳고 그름을 논하고 평하여 단정 짓지는 못합니다. 그러나, 적어도 2017년 12월 제 자신을 만들어준 이 이야기의 감동적인 라스트 컷으로서 미래에 저에게 전하고자 이 졸업논문의 감사의 글을 작성하였습니다.

먼저, 제 대학원 생활의 지도 교수님이신 김대은 교수님께 가장 큰 감사의 인사를 드립니다. 부족한 저에게 배움의 기회를 주셨고, 연구실에 받아주시고 엄격한 가르침으로 연구적, 인간적 자세를 확립하도록 지도해주신 바, 저를 한명의 연구자로 만들어 내보내주신 은혜에 감사드립니다. 또한 제 논문을 심사해주신 이상훈 교수님, 김은태 교수님, 허준 교수님 그리고 명현 교수님께 감사드립니다. 모자란 제 논문에 대해 한계와 발전방향을 제시해주시어 지금의 논문이 나올 수 있었습니다. 언제나 제 인사를 웃는 얼굴로 받아주신 황도식 교수님, 황태원 교수님, 신용준 교수님께 감사드립니다. 친절을 베풀어 주시어 연구실로 인도해주신 심미영 선배님, 연구의 기본을 물려주신 유승은 선배님, 철없는 저로 인해 꾸중도 많이 듣고 함께 겨울 안산을 배회했던 영서, 상욱 형님께 감사드

립니다. 또한 언제나 웃으며 제 말을 들어준 바보 같이 착한 은석이형, 함께 있어 최고로 즐거웠으며 같이한 연구마저 재미었던 재홍이형, 바른생활 코스프레 전문가이자 정당한 폭력가인 세준형 그리고 최고의 친구이자 형이자 인생의 선배로서 존경하는 현구형과 함께한 수많은 나날들을 무엇보다 감사합니다. 그리고 짧은 시간이었지만 함께여서 즐거웠던 서현, 너무나도 똑똑하고 조용했던 선배같은 동생 동현, 나이를 속이고 입학한것 같이 영리하고 근면했던 애늙은이 중보, 유일하게 동일한 연구 주제로 토론을 밤새 주고받던 고집불통이지만 많이 보고싶은 승민과 함께 했던 영광에 감사드립니다. 그리고 이후 연구실에 들어온 학생들인, 누구보다 책임감이 강하고 든든한 동생인 연구실의 버팀목 재현, 불의를 참지못하는 의리파이자 섬세한 도시 남자 슬기, 세상을 향해 도전하는 최고의 실력자이자 인생선배인 승배, 일복의 상징이자 내가 아는 최고의 친구같은 형인 창묵형, 세상 몹쓸 개그 좋아해주고 많은 가르침을 준 제스쳐니스트 혁현, 정 많고 정말 많은 얘기를 나누고 나를 좋아해준 동생 경래, 인생에 선택에서 형으로서 조언과 도움 아끼지 않는 부산 사나이 충현, 마지막에 시각 알고리즘으로 의견을 나눈 차세대 연구실 기둥 민철, 일 때문에 많이 투닥거렸지만 멋진 노력의 상징 상훈, 연구실을 위한 봉사로 점철된 많이 아쉽고 미안한 기억밖에 없는 정훈에게 감사드립니다. 그리고 이번 졸업 동기이자 아는게 정말 많은 실력자이자 감정이 풍부한 예술가 정원(화좀 줄이고)과 오랜 시간동안 연구실에서 큰형님으로 모두를 이끌어준 원기형에게 함께 영국에서의 경치와 파이팅을 생각하며 감사인사를 전합니다. 그리고 저보다 훨씬 아는게 많지만 과시하지 않고 도움을 실천하는 현자 재우와 누구보다 열정가이고 최고의 밤샘메이트이지만 현재 자신과의 싸움중인 만동, 마지막으로 연구실 궂은 일 마다않고 불가사의할 정도의 최선을 다하는 모습으로 인내가 무엇인지 가르쳐준 스승같은 동생 병문에게 특별한 감사를 전합니다.

다음으로 제 불평, 불만을 들어주고 함께 나눠준 친우들에 대해 인사하고자 합니다. 11년 간 인연을 유지하며 학사때부터 대학원 졸업까지 서로를 위한 동지가 되준 윤석형, 현성형, 바울형과 언제나 고마운 창섭, 창진, 종원, 현식, 지현, 조운, 상아들, 소월을 포함한 YRC 전원(너무 많아서 생략)과 못난 형과 놀아주며 인생을 나눈 최고의 인생 동생 환석, 인수, 상민 그리고 안장위의 인연 상윤형, 규근형, 정균형, 창우형, 태준, 민섭 및 팀연세 친구들에게 감사드립니다.

끝으로 긴 시간동안 저로 인해 많은 것을 포기하고 힘들게 기다려주신 가족들과 그동안 울고 웃으며 제 길을 바로 잡아 준 최고의 아군에게 말로는 다 하지 못할 최고의 감사를 전하며 이 글을 마치고자 합니다. 이 창 민 드림

ABSTRACT

Bio-inspired Navigation Based on Egocentric Visual Perception

Changmin Lee

School of Electrical and Electronic Engineering
The Graduate School
Yonsei University

The word, 'navigation', is defined as the study of the control and movement observation process which enables ships or vehicles to move from a certain point to some other designated point. The study of navigation has evolved with the introduction of artificial intelligence and its fusion with GPS, cameras, and distance sensors. Navigation has been used effectively in various industrial fields and in daily life. Recently, visual information has become an important factor, enabling a wider range of experiments. However, visual information also has its own limitations because of the large noise signals captured during data collecting process. Additionally, visual information and removal of the noise it contains requires high complexity in processing algorithms.

It is widely known that visual navigation, which is directly connected to survival, is a common ability of numerous organisms such as ants, bees, and spiders. They follow various behavior patterns such as hunting, exploring, and defending, which are imperative skills for survival. Recently, research involving the complicated visual cortex has revealed that there are certain organizations within the cortex performing location recognition, such as place cells. Also, the snapshot model algorithm is similar to insect navigation, including lower complexities than other engineering algorithms.

In this thesis, various types of bio-inspired (biomimetic) homing methods based on insect behavior are evaluated. First, a new homing navigation method using both holistic range and vision data is evaluated. It is based on the moment function to combine two measurements into a navigation problem. Second a method focusing only on vision using organized masks based on photoreceptor cells is evaluated. The goal is to transform input images into landmark vectors through artificially-generated mask-type photoreceptors. Third is a method that exploits pixel-wise landmark information, which treats each pixel of an image as a landmark called a 'pixel-wise landmark', to emulate homing navigation. Based on a holistic approach, it determines the homing direction by pixel intensity comparison. The performance of each method is measured using various benchmarks and tools, such as angular error, tables, and visualized vector maps in each section. The proposed methods are compared with other existing methods to evaluate the robustness of the methods. Overall experiments are conducted in various environments; custom created environments including various objects, and open datasets such as Vardy's. The overall outputs of the suggested algorithms are summarized and improvements are proposed for future research. At the end of the paper, we organize the appendix for additional explanations.

In summary, this paper, in spired by biomimetics, suggests new visual navigation algorithms by emulating insect homing abilities. Furthermore, an implemented model is validated through simulation processes and robotic experiments.

Key words : bio–inspired homing navigation, snapshot model, holistic visual navigation, visual reception using visual masks, multidimensional moment model, pixel-wise landmark, mathematical convergence

국 문 요 약

시각 공간 인지를 이용한
생체 모방 내비게이션 연구

‘내비게이션’은 배와 같은 이동체에 대해 임의의 지점에서 관찰 정보를 사용하여 목표지점으로 제어하여 이동시키는 기술을 의미한다. 고대의 항해술에서부터 현대의 내비게이션에 이르기 까지, GPS, 시각, 거리, 이동 정보와 같은 다양한 센싱 기술의 발전이 있었으며 이는 인공지능 기술의 발전과 접목되어 무인 자동차와 무인 탐사선 개발까지 이어졌다. 이러한 기술은 일상 생활에서 산업에 이르기까지 직접적으로 적용되고 있으며 최근 적극적으로 시각정보를 활용하고 있다. 그러나 시각이 가진 다양한 정보는 필연적인 민감성을 동반한다. 그러므로 기존의 공학기반 기술들은 시각 정보를 다루는데 추가적인 연산을 사용하고 있으며 이에 대한 복잡도를 무시할 수 없다.

본 연구진은 이러한 시각정보를 단순하면서도 효과적으로 극복하는 곤충의 시각 내비게이션에 주목하였다. 개미, 벌, 거미 등의 곤충들은 다른 귀소성향을 가진 동물에 비하여 적은 뉴런 개수와 낮은 화소의 시각 구조에도 불구하고, 사냥, 탐험, 방어 등의

다양한 상황에서 생존과 직결되는 귀소성향을 성공적으로 보임을 알 수 있다. 최근 생물학적으로 Place cell에 대한 연구를 통해 visual cortex의 복잡한 구조가 종단에서는 위치인식에 사용됨을 일부 밝혀낸 바 있으며, 또한 이러한 visual cortex의 복합 구조를 모방한 CNN과 같은 신경망 연구가 연달아 발표된 바 있다. 본 연구진은 이러한 생체 모방의 강점에 기반하여, 곤충의 귀소 성향에 기반한 생체 모방 알고리즘인 'snapshot model'의 낮은 복잡도에 주목하였다.

본 논문을 통해, 우리는 snapshot model의 생체 모방에 다양한 연구를 조사하여 정리하고 공통적인 문제를 분석하였다. 또한 이에 근거하여 시각 내비게이션의 가능성을 3가지 방법을 통해 개량하고 평가하였다. 첫 번째로, 해당 생체 모방 기법에서 거리정보와 색상정보를 효과적으로 융합하는 방법을 연구하였다. 이는 물리에서 사용되는 모멘트 모델의 개념을 차용하여, 질량 대신 시각 관측 정보를 사용하였으며 이를 통해 임의의 위치에서 집으로의 방향 추정과 수렴을 증명하였다. 두 번째로, 곤충의 시각구조의 기본에 해당되는 'Ommatidia'를 모방하여 시각 전처리가 귀소 연산에 미치는 영향을 연구하였다. 기존의 고화질 이미지를 사용하는 것과는 달리 낮은 수의 화소를 통해 재해석된 시각 정보를 그대로 사용하여 별도의 물체 추출이나 가공없이 높은 정확도의 성능을 구현하였다. 세 번째로, 시각 정보 입력을 곤충의 시각과 같이 각도 별 입력(pixel-wise landmark)을 그대로 사용하였을 때, 귀소 방향 선택에 생기는 문제와 이를 공학적 알고리즘과의 접목에 대하여 연구하였다. 우리는 생체 모방 귀소 네비게이션에서, 전체 시각 정보를 단순비교 하여 귀소 방향으로 결정되는 후보에 대한 수학적 모델을 분석하고 분석된 결과에 대해 공학적 매칭 방법(Dynamic Time Warping)을 적용하여 실제로 사용가능한 수준의 정확도를 얻을 수 있었다. 각 연구에서 우리는 본 분야의 다양한 최신

기존 연구들(MinWarping, COMALV, DID)과 개발된 결과를 비교하였으며, 본 연구진이 만든 환경뿐 아닌 다양한 open dataset과 통계적 지표를 활용하여 결과를 분석하여 해당 방법의 유효성을 보였다. 상기 연구의 결과들을 종합 요약하여 이후 연구의 진행방향과 목표를 설정하였다. 또한 상기 과정에서 발생한 다양한 수학적 모델에 대한 증명을 첨부하였다. 우리는 본 논문을 통하여 생체모방에 기반한 시각 내비게이션 연구를 발전시켜 기존의 연구를 대폭 개량하고 그 성과를 객관적인 실험과 비교를 통해 검증하였다.

핵심되는 말 : 생체 모방 시각 내비게이션, 스냅샷 모델, 전체 이미지 비교, 마스크 기반 시각 인지, 모멘트, 픽셀 랜드마크, 수렴성 증명

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Changmin Lee*)

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In daily life, we frequently hear and use the term ‘navigation’ or a variation of it. Navigation lexically means ”The process or activity of accurately ascertaining one’s position and planning and following a route”. It involves the process of monitoring and controlling the movement of an agent from one place to a goal location. Based on engineering achievements, navigation has been studied from ancient to current times.

Modern navigation systems use various measurements such as GPS, range, and vision. Different to other measurements having high accuracy such as range, vision contains abundant information, but also involves variability to the environment and is difficult to use for navigation purposes, particularly when considering machine navigation. However, robust vision-based algorithms such as object detection, path following, monitoring situation, and environment observation have been developed and applied to various navigation problems in different environments (e.g., land, air, and water). However, these achievements have come at the cost of increased complexity of the algorithms used to perform navigation.

Different from engineering/machine approaches (Davison, 2003), various animals instinctively have a vision-based homing navigation ability tuned to survival behaviors, such as foraging for food. Some insects, such as ants, also have their own simple navigation systems based on measurement using small neurons. In spite of ants’ low calculation capacity, their homing ability to their nests is very robust, enough for survival purposes. Based on ant navigation, many researches have attempted to develop robust navigation algorithms that apply low level calculations to replace the high complexity algorithms currently in use. In this chapter, we introduce a concept of bio-inspired navigation with primary focus on the use of vision and discuss the motivation and objectives of the paper.

1.1 Why Bio-Inspired Models for Navigation?

There have been many cases where common problems have been solved using bio-inspired mechanisms in human history. For example, mimicking of spider webs is used in bulletproof vests and the properties of cocklebur have been mimicked to develop the Velcro fiber. Other natural properties, such as the waterproof property of the lotus leaf, the thrust of a humpback whale, fast (frictionless) skin based on shark's skin, the structure of the quick kingfisher for vehicle design, a streamlined car structure (Mercedes-Benz) based on the trunkfish (boxfish) and architecture applications derived from termite mounds have been exploited by humans. These solutions, mimicking bio-mechanisms, are referred to as 'Biomimetic' solutions, which refers to human-made processes, substances, devices, or systems that imitate nature. The art and science of designing and building biomimetic apparatus is called biomimetics, and is of special interest to researchers in nanotechnology, robotics, artificial intelligence (AI), the medical industry, and the military. Also many biomimetic applications are found in navigation.

There are many examples of animals such as birds that seasonally move over thousands of kilometers. However insects such as ants, having smaller brain capacity than birds, also successfully return to their nest after outside exploration. Detecting the home-bound direction after finishing an outside activity must be an essential ability for the survival of ants building a nest, and implies robust performance. Overall the performance of these homing insects exceeds those of other engineering methods applied in human industry. Researchers have designed bio-inspired algorithms based on these efficient homing phenomena for robotic navigation. For example, there is well-known research regarding ant navigation using pheromones (Willis, 2008) which facilitate one ant to follow another. Similarly, there has been extensive research into methods that accomplish homing navigation using senses, such as vision, olfactory, auditory, odometry, and magnetic (Collett and Collett, 2000; Mather, 1991; Papi, 1990; Rossier et al., 2000; Luschi et al., 1996; Ugolini et al., 2010; Steck et al., 2010; Lent et al., 2010; Gould, 2015). Animals can recognize their location and moving direction using various senses. Among them, many studies have mainly focused on visual cues (Reid et al., 2011; Basten and Mallot, 2010; Wystrach et al., 2012), such as skyline detection, or tracing the sun or moon for navigation. Not limited to ants (Collett and Collett, 2000), there have been many studies of other species such as honeybees (Srinivasan, 2010; Kirchner and Braun, 1994), jellyfish (Garm et al., 2011; Garm and Nilsson, 2014) ,rodents (Etienne and Jeffery, 2004), fiddler crabs (Zeil and Hemmi, 2006), and gerbils (Etienne et al., 1996; Mittelstaedt and Mittelstaedt, 1980).

Early works focused primarily on mapping the environment through measurements. Then a geometric representation of the environment (a map) helps the robot to estimate its current position on the reference map and find its goal direction. However, the large memory required for mapping is too complex to be applied in small insects' navigation. Thus view-based homing navigation has been researched through various bio-inspired algorithms involving simpler processing. These methods were developed to allow a machine to return home by using a stored view of the home location. The basic premise is that insects may use a topological representation of space or may even have no space information stored in their brain.

Among these view-based researches, a snapshot model to model a bee's navigation has been proposed by Cartwright and Collett (Cartwright and Collett, 1983), where a pair of snapshots, one at the current location and one at the nest are compared to determine the homing direction. For homing direction, only two images are sufficient, instead of various images taken along the moving path. In this method, agents move along the direction which decreases the difference between the current view and view from the origin or the nest. If there are visible landmarks available at the two locations, they can help to guide the agent home efficiently. Local visual homing based on this snapshot model is an example of bio-inspired research by using visual cues. As mentioned previously, the method uses only two images (snapshots). One snapshot captures the home position and the other is a snapshot of the current view. Then there must be some differences between the two images that provides some information about the relative location difference. If there is no large distance difference between the home and current positions, then both images are similar, and vice versa. The agent tries to use these differences to find the proper homing direction. As the algorithm exhibits both simplicity and robustness based on bio-inspired bases, other research into methods which attempt to find the accurate direction have been attempted by using various sensors to obtain location information such as color, depth, and optical flow.

The main advantage of a bio-inspired system is its ability to quickly respond to an external stimulus in a simple manner. Animals have specialized sensing mechanisms and studying their properties can assist in the development of compact intelligent systems in robotics. In this respect, previous bio-inspired researches are worth studying, since local visual homing remains a challenging problem. Vision plays a great role in localization, mapping, and navigation. There have been various studies involved with local visual navigation (Huber and Kortenkamp, 2002; Méndez-Polanco et al., 2009; Möller, 2009; Churchill and Vardy, 2013; Denuelle et al., 2015a,b). More advanced algorithms have been developed for simultaneous localization and mapping (SLAM) (Thrun and Montemerlo, 2006; Davison et al., 2004; Dissanayake et al., 2001) or graph models (Hübner, 2005; Jensen et al., 2009). The navigation in our paper concerns a type of navigation based on the snapshot in which the agent plans a path and trajectory on its own to reach the goal point by comparing the current measurement with only one remembered snapshot without any other records such as a geometrical map of the environment.

1.2 Motivation and Objectives

Motivated by the vision-based insect's navigation including both simplicity and robustness, we propose various types of visual navigation systems with various features. The main purpose of this thesis is to develop a robust homing navigation method which can emulate an insect's navigation.

The detailed objectives are as follows:

Various vision-based homing navigation methods. We introduce various navigation models using different features. Based on a snapshot model, we change the measurement function to evaluate different types of navigation. The first method uses a fusion of range and vision by applying the moment function. The second method operates without range data, and is based on the receptors in the retina structure by using bipolar-like masks to convert input vision to landmark information for navigation. The third method uses only an image, and treats each pixel of the image as a landmark to assist in locating the home position through analysis of pixel variation. Through those suggested methods, we study the origin principle of visual navigation.

Performance check through experiments. We also check the performance of the proposed methods through both simulation and experiments, and we compare our methods with several existing navigation methods. These experiments include checking performance to evaluate the robustness of the methods. Each performance is measured using various properties and tools, such as angular error, tables, visualized vector maps, and specialized plots. Overall experiments are conducted in various environments that are described in the Appendix. We use environments we created as well as other open datasets such as Vardy's.

Mathematical convergence proof In addition to the experiments above, we present the convergence of the suggested methods by mathematical proof. With simple assumptions, we evaluate our model convergence and determine any limitations.

1.3 Organization of Dissertation

In Chapter 1, we introduced our research based on the advantages of bio-inspired algorithms including various homing navigation applications. We also specified the motivation and objectives of our thesis. Based on these aspects, we organize this paper as follows.

In Chapter 2, we present background on the existing visual navigation algorithms. Among these, we focus on homing navigation and we categorize the common problems in this field, thus highlighting the goals of this thesis.

In Chapter 3, we propose the moment-based homing navigation algorithm. The method combines both range and visual measurement to achieve robustness. It consists of three major steps: size adjustment for range and vision data, the creation of average landmark vectors in moment space, and locating the homing vector using alignment. Fusion of two measurements facilitates multiple approaches to calculate the homing direction, and we mathematically describe the detailed procedures of the method.

In Chapter 4, based on the receptors in the retina structure, we use visual masks converting input vision to landmark information of navigation without using range data. We use simple Haar-like masks to mimic bipolar cells and convert each matching score into a landmark vector. The method consists of three major steps: generating organized masks according to each rule, image processing with generated masks to build landmark vectors, and landmark vector comparison to detect the homing direction. Based on organized mask generation, we demonstrate the homing abilities of different methods using these masks as receptors of vision input and we check to verify that the collection of multiple images around the home position facilitate accurate detection of the homing direction.

In Chapter 5, we propose the pixel-wise homing navigation algorithm. The method treats each pixel of an image as a landmark and calculates differences in the landmark distribution to detect the homing direction. It consists of three major steps: pixel information extraction with filtering process, generating pixel-wise landmark vectors, and landmark vector comparison for detecting the homing direction. We can locate the homing direction based only on pixel intensities and we check the effect of the matching process by landmark vector comparison. Additional matching is obtained by Dynamic Time Warping (DTW), and improves the accuracy of detecting the homing direction.

Chapter 6, presents our conclusions about the proposed model and discuss future research.

Chapter 2

Background

There are many insects and other animals that use homing navigation based on visual cues. These creatures always explore the outside world over considerably long distances from their nest when for foraging food, guarding the home site, or various other reasons. Thus, being able to return home after finishing a trip is an important requirement for survival. In spite of a low number of neurons in a variety of insects, insects exhibit a variety of navigation skills, including successful homing navigation (Collett, 1996) using various built-in senses. To solve the problems regarding the high complexities of conventional methods to achieve navigation, we focus on various bio-inspired visual navigation methods. The snapshot model (Cartwright and Collett, 1983) is one example of homing navigation by bees. Using only two snapshots from the target and current position, an agent (bee) can easily determine the homing direction by minimizing the differences between two visual cues. The study essentially created a new research area that has expanded with many research contributions. In this section, we introduce the overall navigation techniques and discuss the problems of snapshot-based models as a goal to overcome in our research.

2.1 Conventional Navigation

Artificial intelligence is designed to solve various problems based on various high-level human abilities such as learning, inference, perception, or language comprehension using computer algorithms. Along with the development of hardware techniques, AI algorithms are generally complex and robust. For example, early models applied

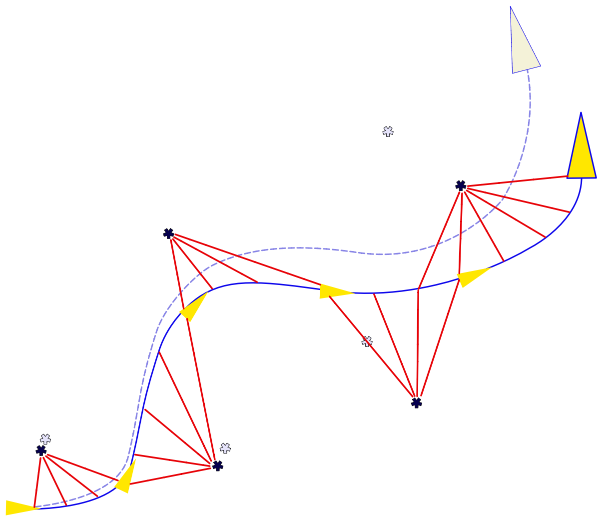


Figure 2.1: The essential SLAM problem. A simultaneous estimate of both robot and landmark locations is required. The true locations are never known or measured directly. Observations are made between true robot location and landmark locations. A single realization of robot trajectory in the FastSLAM algorithm (Durrant-Whyte and Bailey, 2006)

to data classification such as the perceptron (Ruck et al., 1990) cannot solve the XOR and other non-linear classification problems, but later algorithms such as the multi-layer perceptron are able to perform non-linear classification. The Support Vector Machine (SVM) (Furey et al., 2000) is a relatively new algorithm that exhibits higher performance than other traditional algorithms and the recent appearance of the deep learning algorithm (Collobert and Weston, 2008) based on neural networks exhibits much higher performance than SVM. These AI algorithms have been applied in various fields, for example, feature extraction. Researchers use data from various sensors as input and the machine used to run the algorithm produces the output. For example, face recognition uses visual input and the machine runs a search and locates the optimal index if the correct algorithm is applied. If successful, a particular face can be distinguished from other stored faces or from faces not included in learning set. Similar researches are applied in other object detection applications, such as detecting lanes, pedestrians, signal signs, automobiles, etc., using vision input (Cireşan et al., 2012; Baccouche et al., 2011; Ngiam et al., 2011; Ouyang and Wang, 2013; Sun et al., 2014). Similar discussions can be made regarding voice input (Deng et al., 2013).

Recently, these up-to-date AI techniques have been applied to visual navigation research. A number of methods have been developed for the challenging issue of autonomous navigation of mobile robots (Goedemé et al., 2005; Chung et al., 2009; Sasaki et al., 2010; Kang et al., 2012; Chung et al., 2012). The methods vary widely, from landmark-based navigation (Lambrinos et al., 2000; Weber et al., 1999), which is inspired by insect homing behavior, to complex algorithms that require intensive computation and continuous tracking of the information about the environment (Gilg and Schmidt, 1994a; Davison, 2003; Goedemé et al., 2005; Hwang and Song, 2011). Simultaneous Localization And Mapping (SLAM) (Davison et al., 2007) is one of the well-known examples. It is an algorithm that can simultaneously make and update a map (including extension) with localization in each step. Recent SLAM models based on the Kalman filter have been proposed (Hahnel et al., 2003). Since then, many modified versions of SLAM have been suggested to solve the initial Gaussian distribution assumption of the Kalman filter. Gaussian filters (Extended Kalman filter and Unscented Kalman filter) and particle filters (histogram filter, binary Bayes filter and particle filter) have been developed and localization algorithms such as Monte Carlo and other mapping algorithms are attached to SLAM algorithms to created new SLAM versions, such as Fast SLAM (Hahnel et al., 2003; Montemerlo and Thrun, 2007), Graph SLAM (Thrun and Montemerlo, 2006), SEIF SLAM (Eustice et al., 2005). They use various measurements including range, vision, magnetic, accelerometer, and GPS. Especially, range sensors like Velodyne, SICK, and Kinect are the robust range measurement devices which have small error (under 2cm) and they can also provide omnidirectional range data in 2D and 3D. As these measurements are robust, the robots or algorithm using them are also robust.

Various SLAM algorithms are commonly designed to robustly find a landmark and trace it by using probabilistic parameters to fine-tune localization. Figure 2.1 shows the basic principles. There are two conditions. One is estimated and the other is true. Because observed data contains some errors, there is also landmark location information that is not correct. As robot movement also has errors, true locations are never known or measured directly. Observations are made between true robot and landmark locations, resulting in a current location estimation with some error. But, if we record the data and trace it from another trial after some journey, we can correct the prior information. This provides the robustness of the SLAM algorithm.

However, SLAM involves high complexity compared with an insect's navigation. For example, 'Velodyne' is a powerful three-dimensional range sensor that offers vast 3D point cloud data and enables many prototype smart car applications. It also reduces the number of sensors that are attached to an automobile. But an overall price reduction is not achieved because the price of 'Velodyne' is extremely high. There are other SLAM algorithms not using Velodyne, but they also have high complexity that cannot be easily realized using small computers. Similarly, our research using the laser range sensor (Yu and Kim, 2010b) also exhibits good performance. But these devices are very expensive and the algorithms are too complex to apply to the case of insect navigation. Thus, it is necessary to find other solutions that are simple but robust.

2.2 Navigation of Various Animals

An animal is associated with the term 'activity'. Activity can be one or more actions depending on the situation and can represent actions for survival. For example tasks like foraging for something, finding a new nesting place, making a nest, guarding the nest, running away, etc. In essence, most creatures have activities that focus on the nest. Therefore returning to the nest is very important to the survival of each creature and they display robust homing abilities using very small brains. Like a man performing navigation using maps, , the majority of the animals use features they recognize in their navigation, such as the story of 'Hansel and Gretel' who followed bread crumbs.

There have been many algorithms (Anderson, 1977; Wehner and Räber, 1979) that propose navigation methods applicable to robotics. Many have tried to design new high performance bio-inspired visual navigation algorithms incorporating robustness and simplicity. Based on the simplicity of their navigation, there have been many well-known studies dedicated to biomimetics, for example the ant's pheromones (Willis, 2008) to lay and follow a trail. Also animals find their homing direction using various other senses like vision, olfactory, auditory, odometer, and magnetic (Collett and Collett, 2000; Mather, 1991; Luschi et al., 1996; Ugolini et al., 2010; Steck et al., 2010; Lent et al., 2010). There are also many other studies regarding the homing navigation of various species using these senses, such as the Desert Ant (Collett and Collett, 2000), honeybees (Kirchner and Braun, 1994; Srinivasan, 2010), jellyfish (Garm et al., 2011) ,rodents (Etienne and Jeffery, 2004), fiddler crabs (Zeil and Hemmi, 2006), gerbils (Etienne et al., 1996; Mittelstaedt and Mittelstaedt, 1980) and so on.

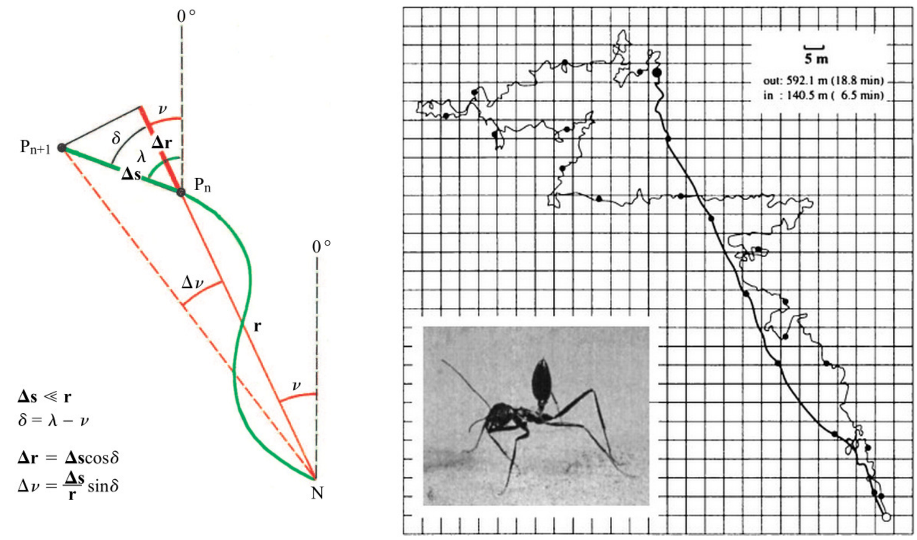
Research into insect behavior continues to date. One example is about alignment of the dung beetle (El Jundi et al., 2016) by using skylight cues. . Also, there have been various studies on ant navigation based on visual cues (Zeil et al., 2014; Srinivasan, 2017; Lent et al., 2013; Schwarz et al., 2017). An interesting study if found in (Collett et al., 2017), which researches the effect of visual cues in backward walking performed by an ant to move collected food. Similar research has been carried out on flying insects. (Stürzl et al., 2016) focuses on how a wasp uses learning flights to gather visual cues around the home position and it takes an arc-shaped returning path when there are visual disturbances. Such bio-inspired homing navigations have generally been divided into the three tasks discussed below.

2.2.1 Path Integration

Path Integration (PI) (Darwin, 1873; Kimchi et al., 2004; Giachetti et al., 1998; Vickerstaff and Merkle, 2012; Wintergerst and Ronacher, 2012; Etienne and Jeffery, 2004) is also known as 'dead reckoning'. This method was first postulated by Darwin in 1873 and described by Murphy (1873). It has been researched in various fields. For example, in the Sahara Desert, the desert ant (Cataglyphis Fortis) performs long-distance journeys for its body size. But this type of ant cannot use the pheromones generally used by other ants because the temperature of the ground in the Sahara Desert is sufficiently high to evaporate pheromones. Also the variable characteristics of the desert can cause visual information to become different in a brief space of time. Then, using Path integration, a desert ant accumulates its movement path from home and it returns home by using the accumulated data. In this manner, it has to find the location and the direction then renew the current location and direction using the previous one in gradual steps from the start position.

$$\begin{aligned} P_{n+1} &= P_n + \delta = (v+, \delta v, r+, \delta r) \\ dir &= tan^{-1}(v+, \delta v, r+, \delta r) \end{aligned} \tag{2.1}$$

Figure above shows the principle of Path Integration. As mentioned above, the agent renews both location and direction using the information above. Then the agent checks the total amount of accumulation at the final moment and finds the reverse direction. Other research involving various animals using PI are (Kimchi et al., 2004; Vickerstaff and Merkle, 2012; Etienne and Jeffery, 2004; Collett and Collett, 2000; Kirchner and Braun, 1994; Mittelstaedt and Mittelstaedt, 1980; Zeil and Layne, 2002) . Also as this method focuses only on the Azimuthal changes, when there are vertical changes, some



(a) (b)

Figure 2.2: The left figure (a) shows the principle of Path Integration (reprinted from (Wehner et al., 1996a)). The position Pn of the animal in relation to the starting point of its foraging excursion (nest) is described by the vector (n,r). The right figure (b) shows the typical foraging trip route of the Saharan ant (Cataglyphis Fortis: reprinted from (Wehner and Wehner, 1990)). Starting at the nest (open circle), the ant searches for food on a random course (thin line) until it finds prey (position marked with the large filled circle). The food is carried back to the nest on an almost straight course (thick line).

errors are introduced in the estimation of distance and direction. Therefore research involving 3-Dimension Path Integration (Wintergerst and Ronacher, 2012) propose methods to overcome the limitation. Others have attempted to prove the fundamental principles of homing navigation using neural structures (Möller et al., 1999; Frost and Mouritsen, 2006; Reppert et al., 2010; Trullier et al., 1997; Issa and Zhang, 2012; Wittmann and Schwegler, 1995; Cheung and Vickerstaff, 2010; Kubie and Fenton, 2009; Kim and Lee, 2011; Haferlach et al., 2007)) and its calculation system (Vickerstaff and Cheung, 2010)

But Path Integration demands a highly accurate measurement model, because PI can accumulate errors and the output of path integration points can lead an agent in a very different direction due to the accumulated errors. Thus, research into mitigating the error accumulation during PI (Samsonovich and McNaughton, 1997; Conklin and Eliasmith, 2005; McNaughton et al., 2006) has taken place. the basic goal is to achieve

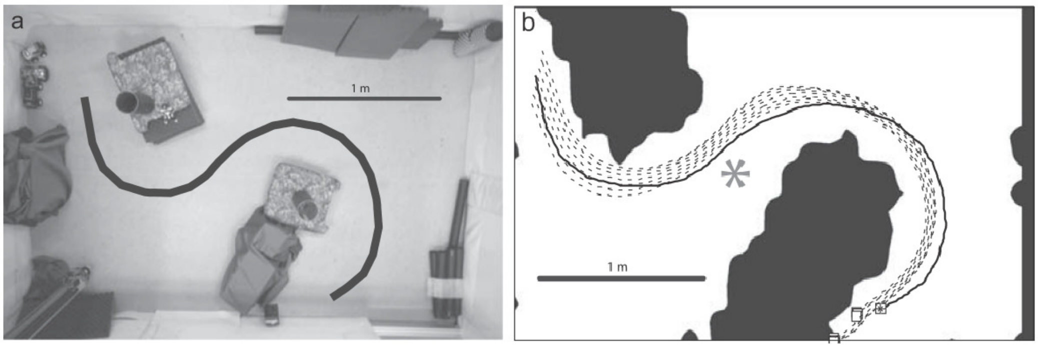


Figure 2.3: Figure shows an example of route following (reprinted from (Baddeley et al., 2011)). Figure (a) shows the workspace as viewed from above. The dark line represents the training route. (b) The output of route following.

the accuracy of an odometer, and some studies have attempted to reach this goal.

The research by (Pan et al., 2011; Dittmar et al., 2010) substitutes visual optical flow information as the odometer measurement. Because path integration and normal visual guidance have different advantages and can supplement each other, some research combines the two methods (Collett, 2012; Merkle and Wehner, 2010) according to the distances involved.

Research on vision-based path integration has taken place recently. For example, (Wehner, 2016) postulates that the ant uses visual cues in early trajectories to build PI. Similar research has also been conducted on various species such as bees (Collett and Graham, 2015), wasp (Collett et al., 2016) and dung beetles (Collett et al., 2016).

2.2.2 Route Following

The Route Following (RF) task uses the images captured during the journey to learn the route. Different to Path Integration, the Route Following method allows the agent to learn the scenes in the route repeatedly. Then the agent follows the learned route using visual cues according to learned output.

Figure 2.3 shows an example of route following which indicates a route is followed by using past learning. Contrary to path integration, this method must invest some time in the learning phase (Schwarz and Cheng, 2010). Therefore if there is some error or omission in the learning phase, the agent cannot carry out its task. And if the agent is in danger and in a location which is not included in learning, then the agent has no reliable direction in which to move. But, regarding learning, this method is very apt for experiments using algorithms that try to produce real models of ants or bees, (Baddeley et al., 2012; Wehner et al., 1996b; Baddeley et al., 2011; Wystrach et al., 2011; Graham et al., 2010; Legge et al., 2010) . A recent study (Wystrach et al., 2012; Reid et al., 2011) applies 'skyline' information (which is the boundary between sky and a landmark in images) to the route following method.

PI and RF can be used in long distances, but at a cost for each method. But PI has the weakness in that the agent must use multiple signals (not only vision) to observe and accumulate its current condition. In the case of route following, if the agent encounters a new path that has not been previously learned, the agent will miss the return path. A common weakness between the two methods is that the agent cannot go to a place unless the agent has fully learned the route to that position.

Other research on vision-based route following has been conducted. For example, research to apply route following of flying insects to flying robots (Gaffin and Brayfield, 2016) uses snapshots from various heights as references and finds the optimal path for route following by calculating image distances. (Gaffin and Brayfield, 2016) presents bio-inspired navigation using the scene familiarity hypothesis (NSFH) in the route following problem.

2.2.3 Local Homing Navigation

The local homing method, as the simplest form in visual navigation, finds the direction from a current position to the target position using only visual information. Therefore this method achieves navigation by only using comparison of visual cues. Following the ant examples, as mentioned above, we have to consider their small brain structure and size with regard to visual information. We assume that the ant has effective data processing and an effective homing navigation method. Thus there have been many attempts to analyze the efficient visual homing navigation of insects. Research can be divided into two categories. One is the case focusing on finding the biological basis of each model (Collett, 1996; Zeil, 2012). The other is the case of applying it into practical navigation applications (Angulo and Godo, 2007).

There are many categories and techniques in this field. The original research of local visual navigation is the snapshot model (Cartwright and Collett, 1983). The snapshot model focuses on the homing navigation of the bee. Using only two visual cues from the target and current position, it can easily find the homing direction. The cue is called a snapshot (SS) from 'home (target)' position and the other one is called the current view (CV). The bee compares those two images to find the differences between them for calculating homing direction. There are many features like simple pixel color, estimated range, height differences, and optical flow available in a snapshot, and the differences of the object's location between both images makes measurements different. The differences can be used to find the homing direction that makes the location of the objects in the current measurement equal to the home's objects. This method follows the premise that the snapshots of both target and current position have common visual information to match each other so the method can be used within a fixed area. Therefore if the distance between snapshots is too far from certain limits, theoretically the agent (bee) cannot find the homing direction. But we can solve the problem by installation of interchanges called milestones (Vardy, 2006) at every limit point. In short, we can develop this method to apply to a wide area by adding only target snapshots which do not require learning or sensor information other than vision. This is then directly related to the short-cut problem (Kubie and Fenton, 2009) which is to find a short cut between two places.

But there is severe problem due to missing range information. Different to other approaches like SLAM, our method is relies only vision without information from other sensors. Therefore, similar research based on the snapshot model has produced various algorithms to mimic range information.

2.3 Classification of Visual Navigation Methods

Before fully discussing the snapshot model, we have to check the overall composition of visual navigation. Figure 2.4 classifies the overall visual navigation techniques divided by Möller into the nodes of a tree, which was introduced by (Möller et al., 2010). As mentioned above, there have been many different techniques and these can be divided into each node of the figure above, which shows a representative flow in this field.

1. Geometrical maps vs topological maps The first norm is the type of the map. The geometrical map is based on geometric information which includes real distances. This

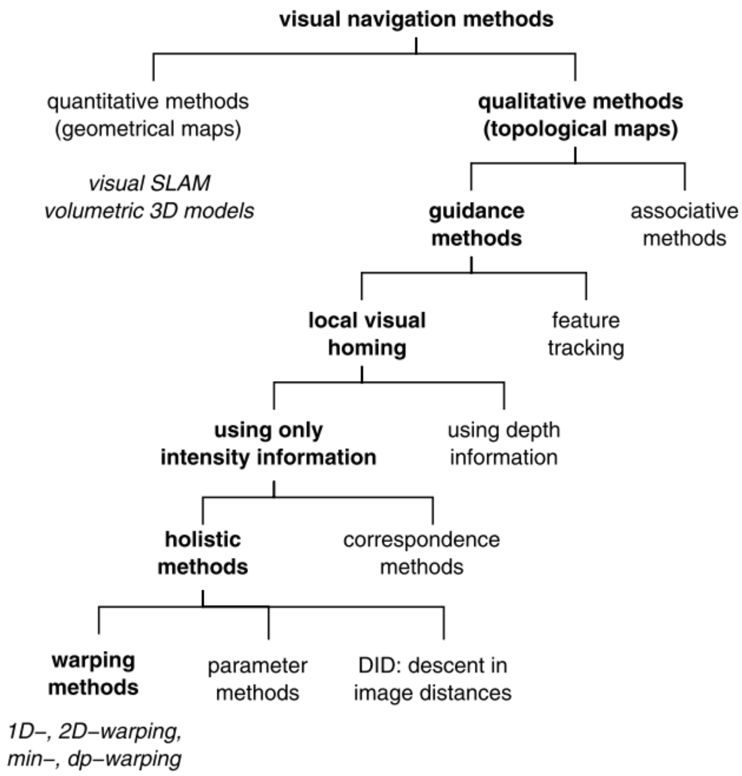


Figure 2.4: Classification of the various techniques of visual navigation into each objective (reprinted from (Möller et al., 2010))

map is essentially the same as maps that people commonly use. In other words, this category includes all the techniques that make (Thrun and Montemerlo, 2006; Davison et al., 2004, 2007; Dissanayake et al., 2001)or refer (Dellaert et al., 1999; Thrun et al., 2001; Wolf et al., 2005; Fox et al., 1998; Burgard et al., 1998; Fox, 1998) to a real map, including the all objects in real distance, and an agent finds its location and position on the map. And the location and position are updated on the map according to the movement of the agent. But because the agent must perform the actual measurement, it must use some type of distance sensor, such as a laser sensor or SONAR, and the amount of calculation is very large due to the complexity of the algorithms required to establish exact position. For this reason, this category is not suitable to our purpose (e.g. simplicity). On the other hand, the topological map is based on the topologies of objects which include relative location between the agent and objects. Then the agent need only find the relative difference of the locations to find the following direction. This method is more suitable to our purpose of simple bioinspired local homing navigation (Meyer and Filliat, 2003; Kuipers and Byun, 1988, 1991; Meyer and Filliat, 2003; Hübner, 2005).

2. Guidance methods vs associative methods Methods based on the associative method (Deng et al., 2011) use vision with an odometer. The odometer largely used in PI research is internal information such as number of steps or turns. In this method, the agent finds the homing direction using vision with accumulated movement information. Even if the agent loses some visual information, it can still perform homing, but will fail to locate the returning route because it finds itself in an unlearned position. Then the robot must re-learn almost all the region to perform successful homing navigation. For this reason, this reason, this method does not suit our purpose (robustness). In contrast, guidance methods use only visual information, where an. agent counts the differences of the feature location and finds the homing direction based on the differences.

3. Local visual homing vs feature tracking Methods based on feature tracking (Gerstmayr et al., 2009) use the correspondences of subsequent image sets (Fiala and Basu, 2004) which include changing features according to sequential visual cues. Then they follow small changes in feature locations. But if the snapshot and current view have a significant time gap between some distances, these large changes of the features are too large to allow the agent to find the homing direction. Therefore we focus on the more generally applicable local visual homing method mentioned in prior section (the guidance method). It divides the entire navigation region into local areas using milestones (Argyros et al., 2005), then these milestones operate like waypoints, allowing successful navigation. In short, if the algorithm allows successful navigation in a small, the cascade combination of local areas allows successful navigation in larger areas. Therefore the snapshot model only uses two visual images and is simple.

4.Using only intensity information vs using depth information A range sensor taking depth information provides powerful information in homing problems. Different to vision, range measurement is very stable with low noise. Therefore using a range sensor makes navigation robust. However, animals cannot visually determine very accurate distance like sensors do and biomimetic research does not generally use range information. One of our models uses the ensemble effect, applying visual cues to accomplish range-based navigation. The other methods use only visual intensity information.

5. Holistic methods vs correspondence methods The correspondence method tries to match the extracted features from images and requires complex algorithms for both extracting and matching them. The holistic method uses entire image matching without extraction or classification of landmark features, and the complexity of the holistic

method is much lower than the correspondence method. We think this holistic approach is the best one for our objective of having both simplicity and robustness.

6 Descent in image Distance (DID) methods vs Parameter methods vs Warping methods The holistic approach can be categorized as a Descent in Distance method, parameter method, or warping method. The differences between them are representatively the choice of landmark and the norm of difference between the landmarks (Collet and Land, 1975). We present the detail of each method in the next section.

2.4 Snapshot Model

Snapshot model (Cartwright and Collett, 1983, 1987; Collett, 1996; Möller, 2000; Wehner et al., 1996b) is one of the biomimetic navigation algorithms applicable to honeybee navigation. As the snapshot method is the progenitor of visual navigation, a number of similar robot visual homing algorithms have been developed based on the snapshot method. The key point of the snapshot model is that the agent remembers only the first and last scenes to accomplish homing navigation. By comparing the current feature information with the first snapshot (home), the agent can determine which direction to move in to reach the target position. This snapshot, which has a panoramic image form, includes the current position in the azimuth direction (Nelson and Aloimonos, 1988). And some researchers (Zeil, 2012; Kohler and Wehner, 2005; Narendra, 2007; Collett, 2010; Reid et al., 2011) insist that the real creatures use panoramic information to navigate.

Then if the distance between home and current is not big and there commonalities between the two snapshots, a simple comparison yields a home vector that points the direction of home. But the length of the home vector is not equal to the real distance between the current position and home, because the simple snapshots have only color pixels without depth information of any landmark. The important part of this method is that if the variation between the snapshots is not severely large, then the agent does not need to remember any scenes between the first and the last to perform successful homing navigation.

The snapshot model has weaknesses regarding the transition of environments. The first is due to changes of the environment caused by an external factor. In this case, there can be severe errors in the cases which have no common features between the snapshots.

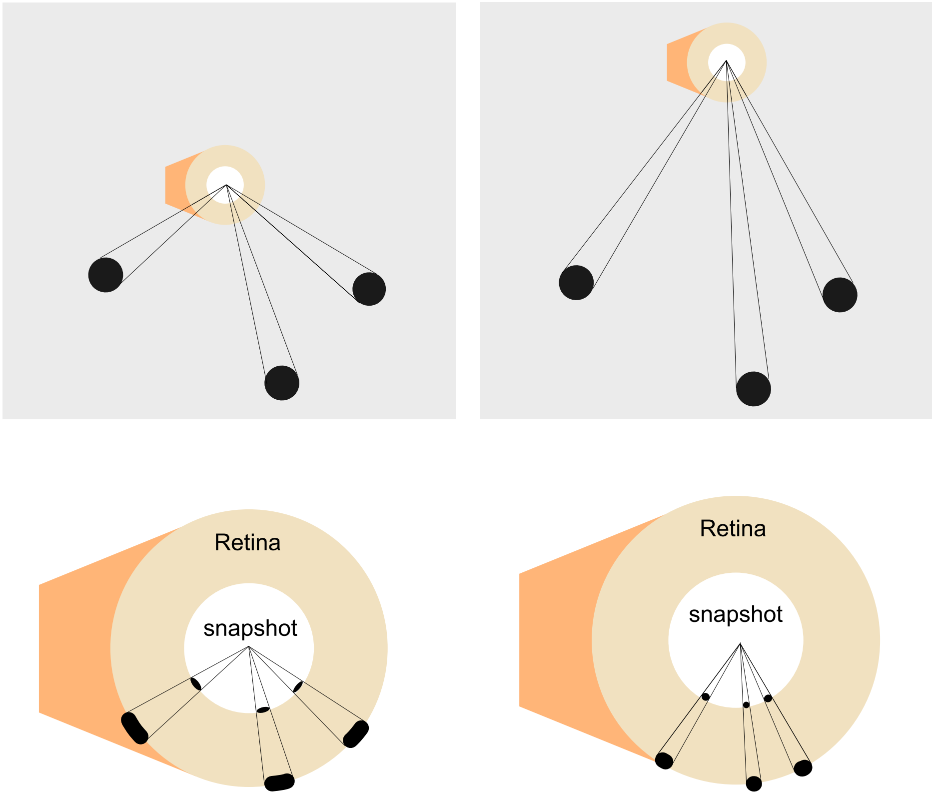
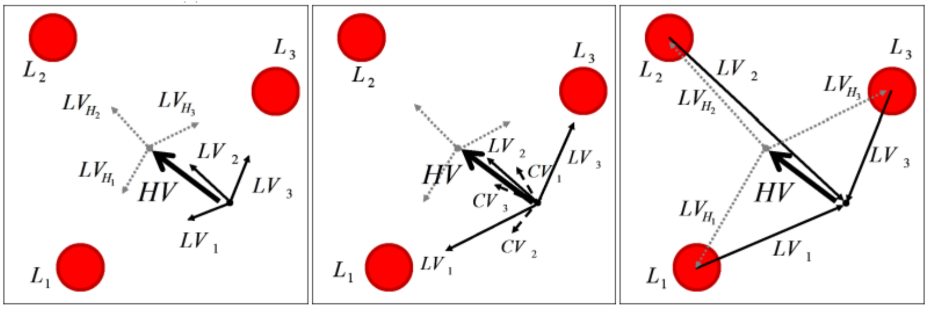


Figure 2.5: Example of the snapshot model (Cartwright and Collett, 1987). Upper left shows that Bee takes a snapshot at the goal. Landmarks are shown by black dots, bee marked as circles with tail at the center position. Outer circle represents retina and inner circle is the captured snapshot. Upper right is the case at another location of Bee that is distant from goal and there are differences in projected image on retina. Lower figure shows enlargement of upper situation at bee's location. Landmarks are represented by thickened black arcs on both outer and inner circles. The differences paired respectively with the closest landmark. Each pairing generates two unit-vectors that point to landmarks without distance information. Radial vectors result from differences in angular sizes of paired landmarks or gaps. At the large central arrow, calculated homing direction is given by the sum of the unit vectors.

It is necessary to assume that there are no large changes in the environment to use the basic snapshot method. The second is relates to changes of the environment caused by movement. If the movement of an agent is small, then there are no big changes between snapshots. But if the agent travels a long distance from home, or the environment has a special arrangement of landmarks that cannot be seen from all positions, then there may be no common features between snapshots. Therefore there must be limits on distances from the home site to use the basic snapshot model. A solution is that the agent must restrict movement to interchanges which have tolerable changes in visual information between them (Vardy, 2006).

But if there are no obstacles or major changes in the environment, the agent can find the common factors between two snapshots and perform successful homing navigation by using the snapshot model. Also, the snapshot model requires low memory which is similar to creatures in nature. In short, the snapshot model represents a simple and clear standard, which has led to a large quantity of related research. Figure 2.5 shows the principle of the snapshot model used in (Cartwright and Collett, 1983). The outer circle shows the panoramic image of the retina and the inner circle shows the saved snapshot image. Figure (a) is the environment at the food site and the black circles show the landmarks. Figure (b) is the retinal image at the food source where the agent saves the snapshot to find this site as the target. The black part shows a portion of the landmark in the image of the retina and the retina image is initially identical to the snapshot. Then, after some movements, the agent wants to find the way to the food source and (c) shows this situation. There are no changes in the landmarks and the agent can check for differences in Figure (d), in the retina image which includes the landmark information between (a) and (c). The agent may then go in the direction which reduces the difference between snapshot and retina images in Figure (d).

These snapshots include information of the surroundings in the azimuth direction (Nelson and Aloimonos, 1988) and the difference in angular distribution can be used as a clue for homing navigation. This simplicity is applied into modified research (Zeil, 2012; Kohler and Wehner, 2005; Narendra, 2007; Collett, 2010; Reid et al., 2011). Convergence is mathematically and experimentally proven and it can also be extended using the milestone method (Goldhoorn et al., 2007; Ramisa et al., 2011). Recently, the snapshot-matching optic flow-based method (OF-SM) (Denuelle et al., 2015a) has been applied to various types of s applications involving aircraft finding their current location compared with a snapshot. We think this snapshot algorithm can be applied to our method. In next the section, we discuss various examples based on local visual homing.



(a) (b) (c)

Figure 2.6: Figure shows the principles of each parameter method (reprinted from [Yu and Kim 2012]). Graphical representations of the homing vector (HV) computation; (a) ALV method, (b) the ACV method, (c) the DELV method . (dotted arrows: landmark vectors at the home location; solid arrows: landmark vectors at the current location, red circles: landmarks).

2.5 Local Visual Homing method - Holistic Method

In holistic methods, there are three categories.

2.5.1 Parameter Methods

The parameter method is based on the snapshot model and the landmark. This method extracts the azimuthal position of the landmark in the snapshots and compares it with current ones. It requires a small amount of memory and uses simple principles to accomplish visual homing. Therefore, basically, the landmark extraction from the image has priority and affects the performance of the algorithm.

2.5.1.1 Average Landmark Vector method (ALV)

The first algorithm is the Average Landmark Vector method (ALV). The ALV method (Lambrinos et al., 2000) is the best-known one based on parameter modeling of local visual navigation methods. There are three steps. The first one is alignment. It basically assumes the compass-like visual compass (Zeil et al., 2003). It needs to build an absolute norm of direction. Insects are known for using both magnetic and polarized light to achieve alignment. The second is landmark extraction. As the vision of an insect has low resolution, it has a rough clustering mechanism that we can easily imitate.

Some algorithms use one pixel as a landmark. After extraction, the final vector is the one matching the landmark vector. The basic premise is to produce unit vectors that point to each landmark with equal length because we cannot know the distance to each landmark without other information (e.g., the eye structure of most insects is not apt to build stereo vision.) Then the sum of the unit vectors is an ALV, and homing direction can be calculated by the subtraction between two ALVs.

$$\begin{aligned}
\overrightarrow{LV_{H_i}} &= (1,\theta_i) \leftrightarrow \overrightarrow{LV_{Ci}} = (1,\alpha_i) \\
\overrightarrow{ALV_H} - \overrightarrow{ALV_C} &= \frac{1}{N}\sum_{i=1}^{N}\overrightarrow{LV_{Hi}} - \frac{1}{N}\sum_{i=1}^{N}\overrightarrow{LV_{Ci}} \\
\overrightarrow{HV} &= \overrightarrow{ALV_C} - \overrightarrow{ALV_H}
\end{aligned} \tag{2.2}$$

The above formula summarizes the ALV algorithm. Landmark vectors are defined as vectors pointing towards landmarks on the retinal image, each of which has a unit distance under the assumption of equal distance. That is, the distance information is ignored but only the angular positions of landmarks are considered. Summing landmark vectors forms the averaged landmark vector called the ALV (Averaged Landmark Vector). Subtracting the ALV at the nest from that at the current position can determine the homing direction. More advanced models have been tested with robotic experiments (Smith et al., 2007; Goldhoorn et al., 2007). Also, the ALV can be combined with invariant visual feature detection (Ramisa et al., 2011). The simplicity of ALV has made it the best-known algorithm for local visual navigation.

But we have to prepare the compass to use this method and the equal distance hypothesis can make the homing navigation worse depending on the location of the landmarks. There has been research conducted using ALV which tries to break the limitations of ALV (Ramisa et al., 2011).

COMALV (Hafner, 2001) is another variant of ALV model. Similar to ALV, it calculates the sum of landmark vectors using the mean of sub-sampled image as length at each position. Different to ALV, it does not require object extraction but it stores the vector projecting to the 'center of mass' in each image. It includes one low pass filter in alignment and make input image into 90 ommatidia outputs and changes them into 90 landmark vectors pointing each angular location. Then final home vector can be calculated through subtraction.

2.5.1.2 Average Correctional Vector method (ACV)

The second algorithm is the Average Correctional Vector method (ACV) based on the ALV. The ACV method (Smith et al., 2007; Weber et al., 1999) is a modified version of ALV. This method also uses the landmarks and finds the differences between them. But it does not use the unit vector and landmark vector set at home. Instead, it uses the angle difference to create the correctional vector at the current position. The direction of the correctional vector is the angle that decreases the angular difference. And the length of the correctional vector is equal to the size of the angle difference. The size of the CV has a function of weight and the corrected angle has a function of revision. This modified method is essentially similar to the snapshot model, and is defined below.

$$\begin{aligned} \left|\overrightarrow{CV}_i\right| &= \left|\theta_i - \alpha_j\right| \\ \angle\overrightarrow{CV}_i &= \begin{cases} \alpha_i + 90^\circ & \text{if } \theta_i < \alpha_i \\ \alpha_i - 90^\circ & \text{if } \theta_i \geq \alpha_i \end{cases} \\ \overrightarrow{HV} &= \sum_{i=1}^{N} \overrightarrow{CV}_i \end{aligned} \tag{2.3}$$

where θ_i is the angle of the i-th landmark in the reference map and α_j is the angle of the j-th landmark in the current position.

2.5.1.3 Distance Estimated Landmark Vector (DELV) method

The Distance-Estimated Landmark Vector (DELV) model is another landmark-based parametric homing navigation algorithm (Yu and Kim, 2010a, 2011b,a, 2012). Different to the other two methods, it does not directly use the angle difference between landmarks. The DELV method focuses on the distance estimation process. But it cannot perform the estimation using only one image. Thus it makes along additional movement along the traveling direction to observe the azimuthal changes using two images at essentially one position. Then it can calculate the estimated distances to each landmark using a trigonometrical function, because the two pictures are aligned by the additional movement, which involves only movement along a straight line. This feature enables it to determine landmark vectors which have angular positions along with estimated distances. The next step is a demonstration of the landmark navigation method without a reference compass but with the distance estimation of landmarks. Replacing the compass information with the landmark arrangement order, our navigation method exhibits successful homing performance. The DELV method was suggested in our previous research (Yu and Kim, 2010b, 2011a) and the effect and results of the quantized distance applied to DELV were described as well (Yu and Kim, 2011b). The DELV method is specified below.

$$\begin{aligned} \overrightarrow{LV_{H_i}} &= (R_i, \theta_i) \text{ and } \overrightarrow{LV_i} = (d_i, \alpha_i) \\ \overrightarrow{PV_i} &= \overrightarrow{LV_j} - \overrightarrow{LV_{H_i}} \\ \overrightarrow{HV} &= \frac{1}{N}\sum_{i=1}^{N} \overrightarrow{PV_i} \end{aligned} \tag{2.4}$$

DELV can use the estimated distance of landmarks, which makes the direction more accurate. And it is well-fitted to the landmark arrangement from the calculation view, which helps the operation. But the additional movement and two feature extractions at every position cause the algorithm to be slow. Also, there can be double measurement error at each measurement (see (Yu and Kim, 2011b) for a research study that uses quantized information to reduce the error in measurement). Currently, research to improve DELV without using feature extraction and additional movement by using color information.

The DELV model was initially tested using vision sensors, so an additional distance estimation process is required for selected landmarks (Yu and Kim, 2011b,a). Based on the visual information, variations in the landmark positions between snapshots obtained over a sequence of robot positions facilitates the computation of the landmark distance. By exploiting the information from a depth sensor, the distance information to each landmark can be given along with the landmark direction, which provides a robust navigation method.

2.5.1.4 Recent research trends

The ALV based methods basically need object extraction to find the angular location of a landmark. However, finding objects from a simple image is not an easy problem. Therefore early research used designated markers and used various feature point algorithms. This type of method can easily define extracted feature points as objects and also reduce occlusion effects based on feature matching. Improvements have been proposed recently, for example, research (Liu et al., 2013; Gupta et al., 2017a) using ALV with SIFT. Based on the snapshot assumption, it extracts SIFT features from two images and filter the outliers out. Then remaining features are connected to each other and the homing direction is easily determined from the parameter method. Similarly, (Strydom et al., 2016) uses optical flows based on the pattern of motion of feature points in a visual cue originating from the relative motion of an observer. Also, though not representing simple calculation, another approaches using repetition to produce strong matching such as (Ma et al., 2017), which uses locality preserving matching (LPM) with repetition matching. So, parameter-based methods have been widely researched in robotics.

2.5.2 Warping Methods

Another local homing navigation model is the warping method (Franz et al., 1998; Franz, 1999; Möller et al., 2010; Zhu et al., 2015). Different from the other approaches, the warping method estimates all possible changes in image pixels depending on every possible movement of a mobile agent, and also calculates the effect of each movement to each pixel. Therefore it distorts the current snapshot image in order to best match the target snapshot image. The agent predicts a new image for every possible direction that it could move in from the current location by warping the snapshot image. Each warped image is compared to the target image by searching for the one with the smallest discrepancy. The warped image with the smallest difference indicates the direction in which to move in order to reach the target location. The performance of the predictive warping method is largely affected by the characteristics of the environment and the existence of a reference compass; however, the method is simple and does not require any pre-processing of feature extraction in the snapshot image. This Warping method is also based on the snapshot model but finds the optimal pixel movement among the possible pixel matching. Then there are four algorithms which use different pixel matching, discussed in the following sections. In short, they calculate all the cases with variation of the compass, pixel distance, and angular difference of pixels. This method shows robustness in accuracy but requires high computation.

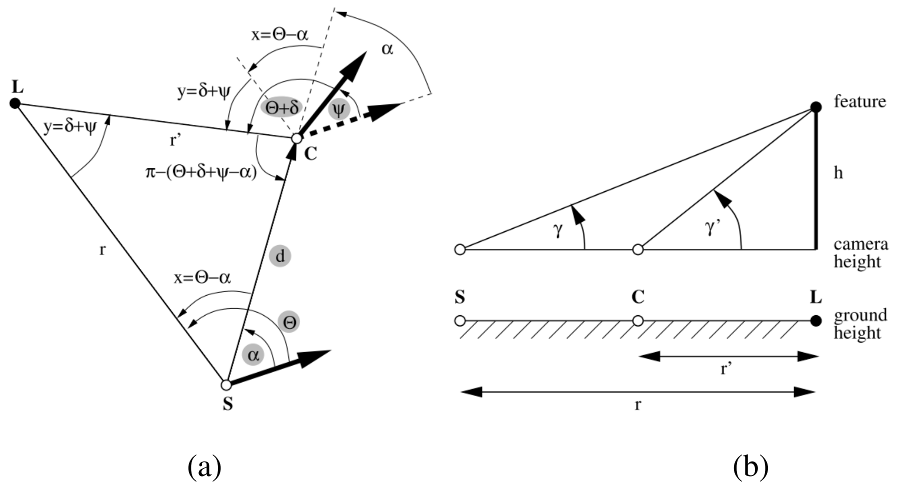


Figure 2.7: Derivation of the warping equations (left: top view, right: side view). (a) A robot moves from a snapshot location S to a current location C, and indicates the difference of the azimuthal plane for 1-D warping. (b) shows the elevation changes for the two locations. It is used in vertical scaling for 2-D warping (reprinted from (Möller et al., 2010; Möller, 2009)).

2.5.2.1 1D-Warping

The first is 1D-Warping (Franz et al., 1998; Franz, 1999; Hong et al., 1992). This method is the basic form of the warping method. In this part, assume that the presence of the compass is to align the images and the agent focuses on the single pixels in horizontal rings. The equal distance hypothesis for pixels in the images is then applicable. Then the homing direction can be determined by using both angle difference and relative distance with the warping function as a trigonometric problem. The angle difference is the turning angle between home and current positions and the relative distance is the ratio between distance to a landmark and current position.

But this method considers only horizontal pixels. There are losses of matching pixels by any change in the vertical direction. So the performance of this method is lower than others.

2.5.2.2 2D-Warping, min-Warping

The second is 2D-Warping and the third is min-Warping (Möller et al., 2010; Labrosse, 2007; Stürzl and Möller, 2007). Different to the 1D case, this method uses 2-Dimensional information. In other words, it refers to both azimuthal and vertical changes. For the

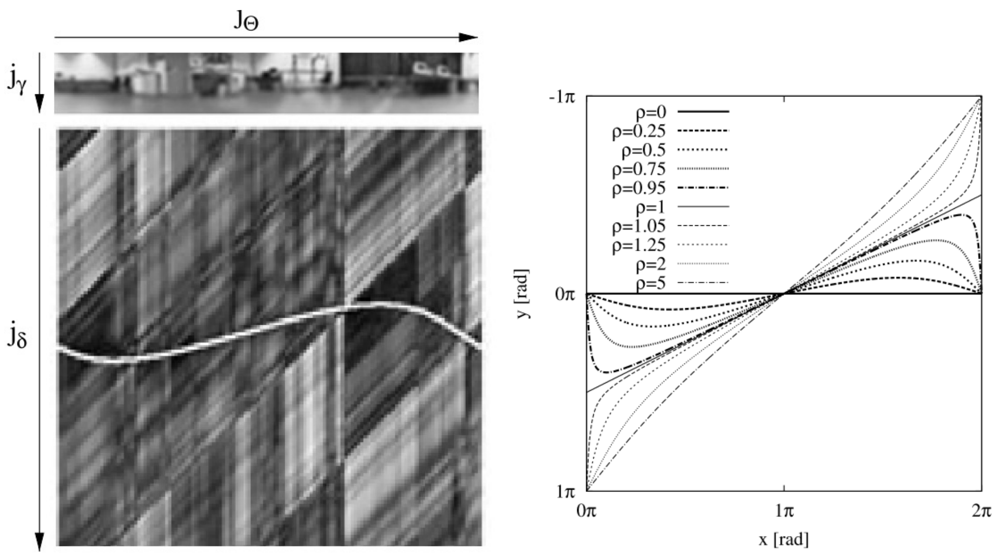


Figure 2.8: Scale plane example (reprinted from (Möller et al., 2010)).

azimuthal changes, it uses a warping function which is equivalent to the 1D-warping's one. And, for the vertical direction, the panorama image with magnification is scaled using the vertical scale factor in Figure 2.8. (b). Then the scale planes for all possible cases are derived and summed to produce the optimal warp curve. Finally the homing direction and angle difference are determined from the optimal point of the warp curve.

Different to 1D-warping, this method determines the optimal direction from all possible positions, so it does not need a compass. The use of vertical components improves robustness. but the increase of dimensions also increases the calculation requirements. Calculation time is also needed to determine the compass value, further adding to the required computation time. In short, 2-D warping is fairly slow.

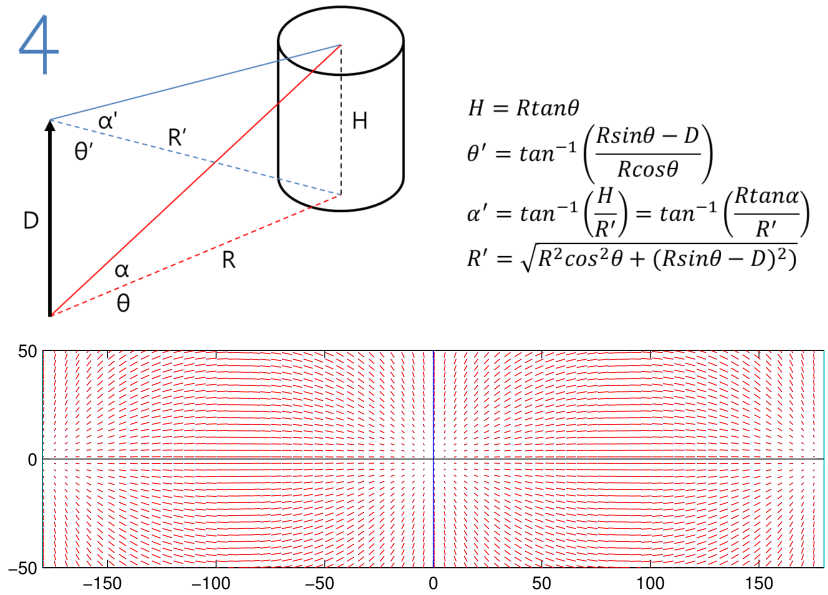
The third method (min-warping), based on a holistic local visual homing method, compares image columns pixel-wise to consider distortions caused by translations. In first phase, it computes all pairwise distances between image columns and tries to estimate the relative orientation between images, similar to the visual compass. By using various scale planes making vertical distortions, it finds the effect of distance to an object changes. Also it calculates all the matched errors including both scale variation and horizontally shifted combinations in scale plane stack. Then, it collect minimum cases from entire plane stack and optimal path is selected to finding homing direction. However, above technique requires a lot of 2D-warping variables and it includes some acceleration processes with partially lifting equal-distance assumption. The Min-Warping uses the computation decreases like compass estimation, pre-calculated warp and plane template, early termination in searching within scale-plane. In this part, the agent finds the angle difference between home and current positions and extracts the 30 percent section that includes the expected angle of the visual compass. This 30 percent is a safety factor to allow for the error of the visual compass. 2D-warping is then performed only in that 30 percent expected angle section, and reduces calculation requirements by 70 percent. In this part, three warping methods (1D, 2D,and min-warping) that use equal distance hypothesis are presented. A fourth method not discussed attempts to solve the equal distance hypothesis problem, and is called Free 2D-warping (dp-warping).

2.5.2.3 Recent research trends

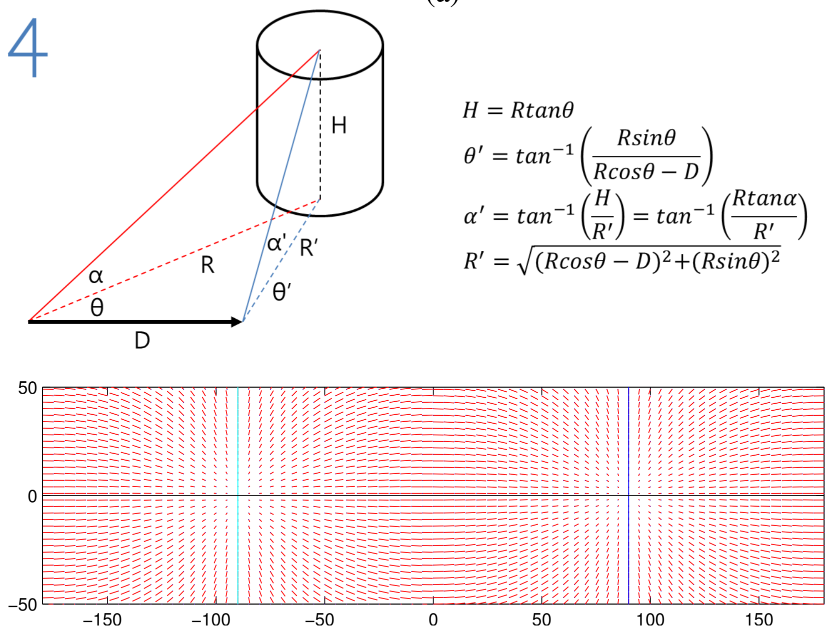
As the warping model is based on pixel matching in various pose changes, it has been widely researched in robotic applications. Typically, min-warping research dominates the field (Möller, 2012, 2016b,a). For example, some studies (Fleer and Möller, 2017; Zhu et al., 2015) use various features such as SURF and SIFT applied to robotic experiments based on warping. Others (Vardy and Möller, 2005) attempt to combine warping with optical flow or particle filters with min-warping to localize a cleaning bot on the topological map the bot builds. These warping-based approaches have greater robustness to illumination problems (Möller, 2016a; Möller et al., 2014) compared with other models and apply warping models into various robotic navigation applications.

2.5.3 Descent in Image Distance (DID) Methods

The Descent in Image Distance (DID) method extends the concept of the warping method. The DID method was introduced by Zeil et al. and was investigated from various perspectives (Zeil et al., 2003; Möller and Vardy, 2006). It finds homing direction by comparing multiple reference images gathered from neighboring positions around the home position with a current visual cue. As the original one manually chooses the most similar image with the home one, in this paper, we use revised version by only using three fixed reference images (H_X, H_Y and H_0) around the home location (x, y) The images H_X and H_Y are collected at $(x+\delta x, y)$ and $(x, y+\delta y)$, respectively. Then it calculates image differences between current one and references and finds ratio them to find homing direction.



(a)



(b)

Figure 2.9: Flow template output based on warping modeling. (a) the y-axis moving case, (b) the x-axis moving case. The two colored vertical lines represent two stationary positions.

While the one-dimensional warping method only searches for the minimum difference from the home image among the candidate images, the DID method monitors variations of the image differences between a pair of locations. The difference between snapshots decreases as the current location approaches the home point. By applying a gradient descent method, the navigation algorithm successfully finds the direction to the goal location (Stürzl and Zeil, 2007; Labrosse, 2007). In short, the DID method is the snapshot-based method using pixel differences. DID requires the compass to align images. Then we assume that the agent has a compass sensor and all of the images are aligned. The agent then compares the snapshot image pixel value at home with the current image pixel value. In other words, DID counts the difference of the entire pixel values between two panoramas and finds the optimum direction based the on lowest difference.

The basic DID method involves taking the new snapshot which is some distance from the current location (this idea is based on the visual compass (Zeil et al., 2003)). So, first the agent takes a snapshot at home and saves it. At some different location, the agent moves a fixed distance along the measurement angle takes another snapshot and measures the pixel differences between the position and home. The pixel difference is calculated by the subtraction between all pixels in both images. After returning to the current location, the agent repeats the process using different locations along different angles. The optimal direction is the direction in which the difference is lower than the current position and the reduction amount is largest. The calculation time is very large, so there have been studies attempting to reduce this problem.

2.5.3.1 Gradient-based Descent in image Distance method

This method (Möller and Vardy, 2006) is the basic application of DID history. As mentioned above, we have to check the direction by visiting in the vicinity of the home position, which produces a great deal of wasted computations in order to find the home direction. The key idea of the gradient-based method is to reduce the number of movements. In short, it tries to find the homing direction using only the home snapshot and three images (the original method used a number of movements equal to 360/N. N is the angle resolution). The first image is on the head direction (North), and the other is perpendicular to the head direction (East), the last one is at the current position. The amount of movement is greatly reduced and the four images including the home snapshot are aligned by the compass. The next step is calculation of pixel differences between home and the other images. The pixel difference between the home snapshot and an image can represent the transition of each direction. In short, the output of the heading direction image shows the difference in heading direction and the perpendicular image captures the perpendicular changes. The homing direction is then determined by calculating the ratio between them. In some cases, two images on the heading direction and two on the perpendicular direction captured by back and forth movement are used (total of four images excluding the current one)

$$\theta_{H,DID} = \tan^{-1}\frac{\sum_{i,j}|H_Y(i,j)-C(i,j)| - \sum_{i,j}|H(i,j)-C(i,j)|}{\sum_{i,j}|H_X(i,j)-C(i,j)| - \sum_{i,j}|H(i,j)-C(i,j)|} \tag{2.5}$$

2.5.3.2 Newton Descent in image Distance method

Using the gradient-based method reduces the amount of calculation, but there must be loss of information and some error as trade-offs. Thus, it is necessary to reinforce this method. The Newton method is applied to this algorithm. In short, we add some references in diagonal differences. The amount of computation is slightly increased, but the accuracy is improved. Some versions use two images on head directions, two on perpendicular directions, and four on diagonal positions taken in a back and forth manner (total of 8 images excluding the current location). This is called the Newton-based DID method.

The Gradient-based DID and Newton-based DID methods innovatively reduce the amount of calculation. But there is still some waste of time in moving (to take the additional images) in comparison with other local homing navigation methods.

2.5.3.3 Matched Filter

To solve the additional moving problem, a Matched Filter (MF) is added to these algorithms. If the objects in the environment have enough distance within a certain range from each other, we can approximately predict the pixel movement in additional images without moving. This is the key idea of the MF.

Figure above shows the principle of MF. The flow template is the output of the matched filter, which represents the expected pixel flow in that direction. The moving distance and the distribution of the objects are not known in the experiment's field. But if we decide on a moving distance and perform some pre-experiments to obtain a suitable flow template which is very similar to the real image, then the predicted image is also reliable. Therefore we repeat the experiment to obtain a credible matched filter. Then the DID method using predicted Image by MF is applied without additional movement.

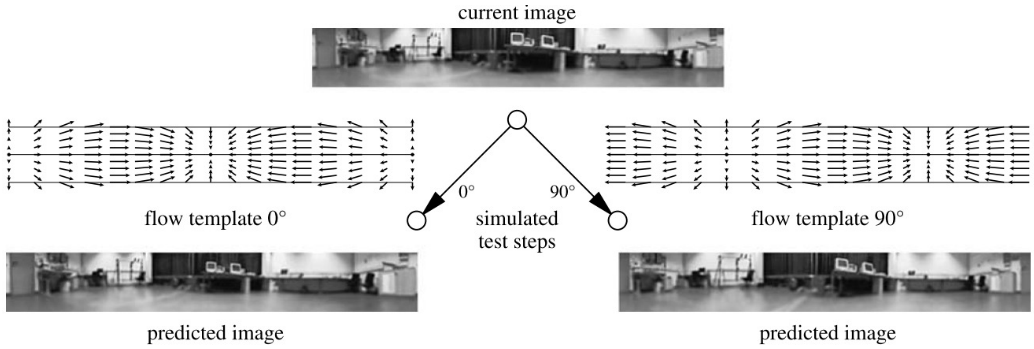


Figure 2.10: Figure shows the principle of MF. Two estimated images are made by applying two translational flow templates for perpendicular movement directions to the current image, (reprinted from (Möller et al., 2007))

These flow templates from the MF include some error caused by the arrangement of the objects but if that error is small, MF is a viable addition to DID and reduces the computation time.

2.5.3.4 Gradient-based Matched Filter DID and Newton-based Matched Filter DID

The predicted image for an arbitrary direction can be obtained by pre-experiment fitting using MF. Then we take into account the additional movement and predict the image at that point to easily replace the additional movements required by the basic DID method. Then Gradient-based Matched Filter DID which includes two (or four) predicted images or Newton-based Matched Filter DID, which includes 4 (or 8) predicted images can be applied without the agent having to move.

Because the expectation processes of MF yield an approximation, errors can occur. Therefore we have to use these methods when the experiment environment is sufficiently large and the MF has a high confidence level for the prediction.

The DID model has been studied in various roles over the years. Homing in scale space (Churchill and Vardy, 2008) involves research using correspondences between SIFT features to analyze the resulting flow field to determine the direction of movement. It is mathematically justified by other research (Churchill and Vardy, 2013).

2.5.3.5 Recent research trends

The DID based methods have suggested linear proportionality between physical distance and image distance. Therefore it has been widely researched regarding visual navigation problems using scene familiarity for route detection. Similar to the warping-based method, there has been much additional research into DID-based applications. Below are a few examples. (Horst and Möller, 2017) tried to solve the illumination problem by using image similarities. (Denuelle et al., 2015b) uses Image Coordinates Extrapolation (ICE) in UAV applications. (Hamaoui, 2017) uses polarized light as input and finds homing direction by applying a gradient DID model (Wystrach et al., 2016) divides input measurements into two parts from tussock and trees and finds the proper direction based on similarity itself. (Stürzl et al., 2015) uses multiple images to render an imaginary 3D environment to expand the experiment area. Also, in the route following problem, (Berenguer et al., 2015) makes reference to images on the grid of a local map and finds the proper location by image familiarity using Radon transforms.

Based on scene familiarity, DID can be compared with the recent discovery of place cells (O'Keefe and Burgess, 1996) and grid cells (Moser et al., 2008; Derdikman and Knierim, 2014), which are place-modulated neurons in an animal's brain. The discovery of their existence supports the idea that homing navigation in animals is supported at the neuron level regarding localization of topological and geometrical features. These neurons appear to support a complex firing system that can define locations within a series of learning images. The cell structure is thought to produce a grid shape covering the entire surface of a learned two-dimensional environment. Thus it appears that DID-based methods and grid cell-based navigation have similarities and relevant research has been conducted in robotic navigation problems using route learning (Fan et al., 2017; Thrun and Bücken, 1996).

2.6 Problems of Snapshot Model

The holistic matching algorithms discussed above share some common issues. First is the localization problem caused by the absence of distance information. Second is the occlusion problem that may result from using only two images. Third is the selection of the view used for visual cue input. Fourth is alignment between the home snapshot and the current view. Fifth is how to select proper landmark features. These five problems must be considered in local visual navigation. The following sections discuss the details of these problems and potential solutions to them.

2.6.1 Problem 1: Localization

The localization issue is a challenging issue, even in modern Simultaneous Localization and Mapping (SLAM) technology. Information about the environment can also be obtained from several types of sensors. The global localization map can be obtained with a global positioning system (GPS) and a laser-ranging sensor. The laser range sensor can also be applied to human-robot interactions (Chung et al., 2012) or industrial applications (Kim et al., 2013). In addition to the distance information about the environment, the Monte Carlo approach with an elevation map of objects or three-dimensional features commonly observed from the air and ground can be used to provide better localization information (Kwon et al., 2010; Kwon and Song, 2011a).

Without a range sensor, visual information has been most widely applied to navigation algorithms, since it can provide much information about the position, brightness, color, and shape of objects. Many navigation algorithms have been developed that apply feature extraction using vision (Zhu et al., 2014; Aranda et al., 2013, 2017), often with a complex filtering process (Chen, 2012). Even simple landmark-based navigation methods, including the ALV model, have been applied to a mobile robot with vision sensors, and a feature extraction process to identify landmarks is also required (Zhu et al., 2014). In contrast, the DID method is a holistic approach for comparing a pair of snapshot images pixel by pixel to calculate the image distance (Zeil et al., 2003; Möller and Vardy, 2006; Labrosse, 2007). It requires no feature extraction process; however, computing the gradient of the image distance is needed to determine the homing direction. These methods exhibit successful homing navigation but lose some accuracy and cannot always determine the optimal path over long distances, so more advanced models have been developed to overcome the distance factor. Different to intensity models, using a range sensor with vision can easily provide location information. other research adds range information to intensity model. (Yu and Kim, 2012; Lee et al., 2017) applies a method that uses a range sensor to easily achieve high performance in both alignment and accuracy of homing navigation.

Then there are methods adding high complexity such as additional image processes, collecting additional images or, introducing additional calculation. For example, segmentation, feature extraction, or full search based matching algorithms are the typical ones that have been discussed previously. These methods involve extensive calculations and normally unsuitable for real-world applications. Thus view-based homing navigation has been researched through various bio-inspired algorithms involving simpler processing, such as the focus of our paper, which is the snapshot model.

2.6.2 Problem 2: Occlusion Problem

Landmark occlusion refers to missing or an obstructed view of one or more landmarks in a location image. Basically, the total number of landmarks expected is not present in an image, and the agent cannot perform adequate matching. If there are four landmarks and the agent can find only three of them due to occlusion in the image, then the agent lost 25 percent of the available visual information. Therefore landmark occlusion can cause severe navigation errors.

In the Möller's classification, other techniques that do not apply not local homing navigation apply compensating tools, such as estimation, other sensors, or sequence tracking, and other methods such as DID and Warping do not extract the landmarks. The occlusion problem is thus specific to parameter methods based on the snapshot model. Figure below shows occlusion effects on different algorithms.

When there is no occlusion, the ALV shows the highest accuracy. Then we can say that the alternative distances of DELV and ACV are not perfect. But if there are occlusions, the ALV exhibits a drastic decrease in performance. Different to ALV, the DELV method is robust to occlusion. Therefore the effect of occlusion can be different depending on the algorithm and the environment. Also, there have been researches trying to filter the occlusion effect by using image reconstruction (Lee and Kim, 2015).

2.6.3 Problem 3: Field of View Problem

At first, we assume that we use the snapshot model in a new algorithm. The panorama image as the input visual cue has all information about the surroundings, such as floor, sky, landmarks, walls, part of the robot, etc. The question then becomes whether all parts of the panorama picture are really necessary, such as the floor, the omni-camera platform, or ceiling. This is called the field of view (FOV) problem.

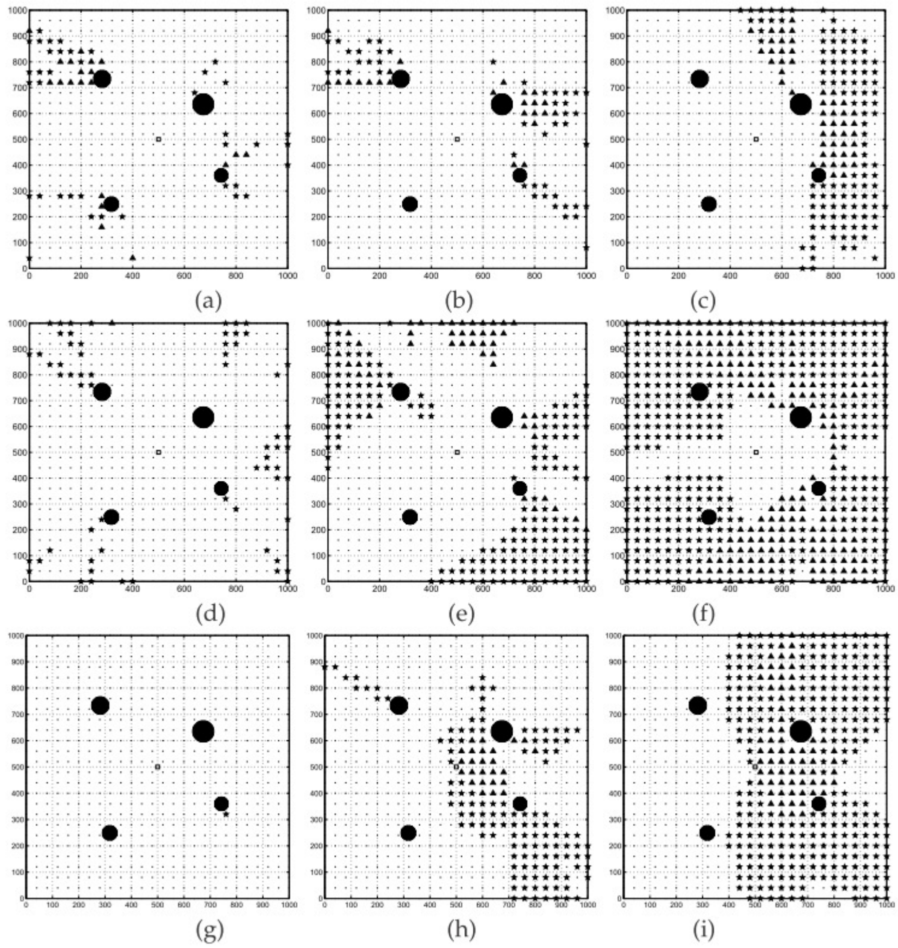


Figure 2.11: Graphs showing error points as the number of occluded landmarks increases from (a),(d), (g) zero to (b), (e), (h) one and (c), (f), (i) two using the (a-c) DELV method, (d-f) ACV method, (g-i) the ALV method. (dot: less than 45, star: between 45 and 90, and triangle: greater than 90, reprinted from [Yu and Kim 2012])

Figure above shows the effect of field of view. Though this figure is an example, there are some differences introduced to the visual information for differences in vertical position. The variance can affect the amount of information needed for navigation. Therefore we have to consider what FOVs to use in navigation for the snapshot model and what FOVs do other techniques use.

2.6.3.1 Skyline

The first method uses the skyline. The skyline is the boundary between sky and other features, including the ground or landmarks. The boundary is located based on color difference between the sky and other objects. Then the amount of data is reduced to

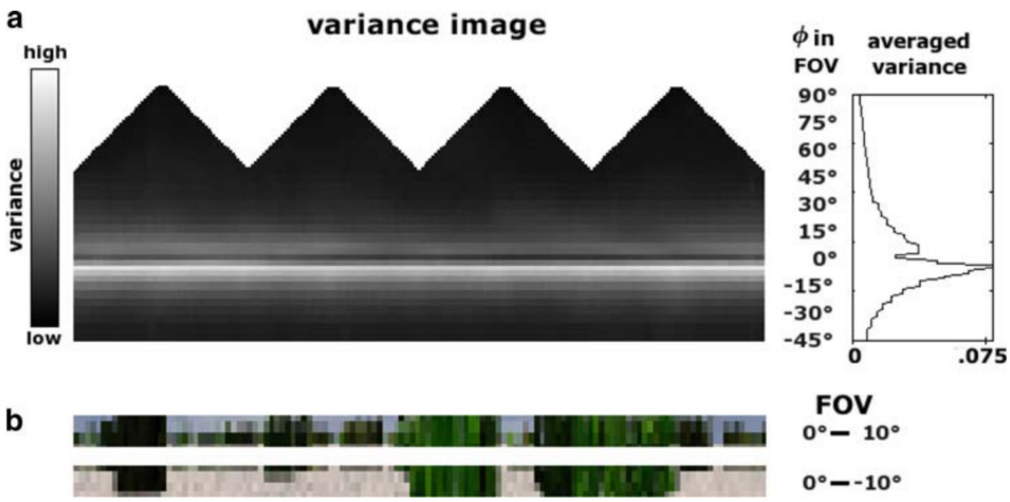


Figure 2.12: a: Local variance of intensity values (range 0–1.0) in panoramic images of a virtual environment. All images are oriented in the fixed viewing direction. b: Two (enlarged) examples of input images with vertical FOVs of 0 degrees to 10 degrees and 0 degrees to 10 degrees providing an impression of the image areas with the highest variance (reprinted from (Basten and Mallot, 2010))

the number of angle resolutions. The skyline includes any exceptional landmarks such as giant trees or mountains and can facilitate robust performance over long distances (Wystrach et al., 2012; Reid et al., 2011). But prominent landmarks do not yield noticeable differences in short range travel while closer objects produce large differences. So this method has weakness related to short distances. Therefore we have to use this model to appropriate environment.

In spite of this weakness, there has been much research regarding insects using the skyline for navigation (Graham and Philippides, 2017).

2.6.3.2 Holistic view

The holistic view is a verbatim method which uses the all pixels in the panorama image. This method is used for pixel matching. All pixels between images are matched to obtain reliable data because the method does not use any designated pixel from feature extraction. Nevertheless there are still some unessential areas in the image. So, generally, the method cuts out some area, such as the floor or sky from each image. But the amount of cutting can influence the accuracy of the output.

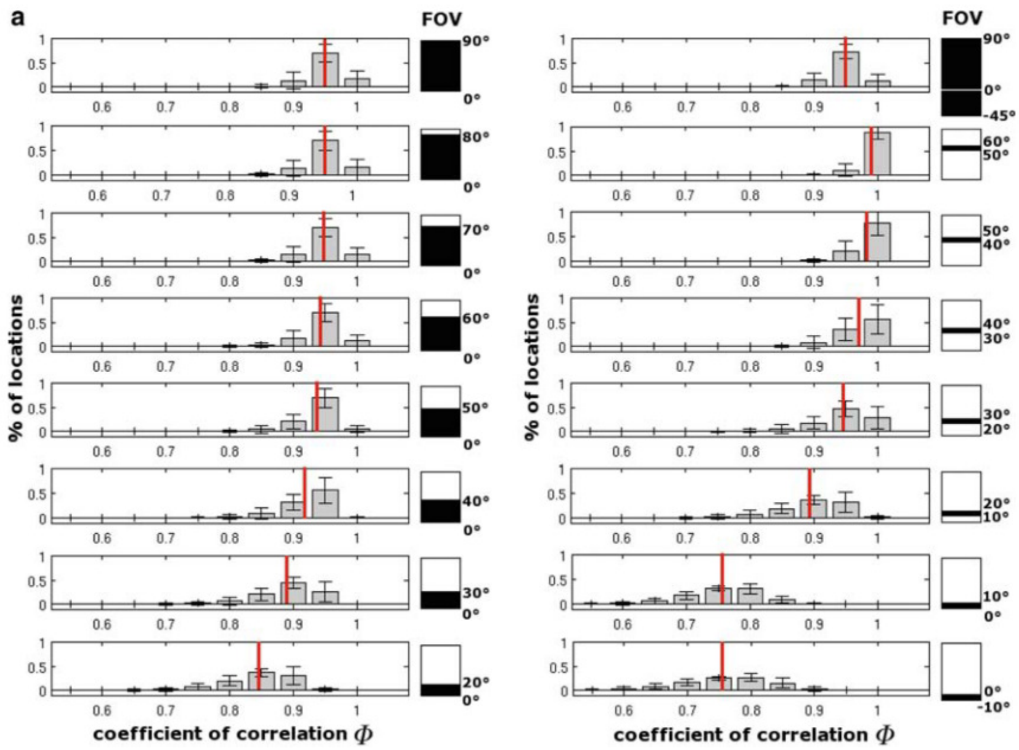


Figure 2.13: Histograms of code correlation; The heavy red line indicates the overall mean correlation. (a) Intensity code. The different panels show correlations obtained with various vertical FOVs (reprinted from (Basten and Mallot, 2010))

Also, another study (Buehlmann et al., 2016) regarding visual navigation of an ant argues that the ant uses not only the skyline, but also shapes. The basic premise is to designate the proper portion of reference in an image. The larger the portion, the more computation is needed. If the portion is too small, then significant information is lost. So a suitable size of the vertical section of the panorama images must be selected.

2.6.3.3 Horizontal Ring

The horizontal ring is the line in the panorama image which includes all information about a landmark location on the ground viewed from the horizon. Thus, in this method, the agent picks up a line of pixels in the horizontal ring. And techniques which are used in almost all of the parameter and warping methods use this horizontal ring for feature extraction. But when we use only one line of pixels along the horizon, there can be some problems missing landmarks for some reason, such as distortion caused by light. Thus it is best to use a set of pixel lines around the horizontal line rather than a single line.

2.6.4 Problem 4: Alignment Problem

The presence of the compass in navigation is an important issue. In this process, Compass directions can be interpreted as an absolute standard. A snapshot is taken, which is represented by visual information, and the locations of landmarks are determined. If the compass sensor provides an absolute norm of direction and there are no major changes in the environment, the order of the landmarks in the snapshot is almost the same for different snapshot taken at different locations and we can easily observe the variation of landmark locations.

But if we do not have a compass, we must try to match the sequence of landmarks or correct the skewed angle between them first as uncertainty variables in our calculations. Thus the many applications have used an external magnetic sensor as the compass, but there have been other various approaches that try to omit the compass and find the direction using a compass sensor. We discuss some examples below.

2.6.4.1 Using Compass sensor

The first method just uses a compass. This is the most common way to fit the rotation of the current image to home. The existence of the compass provides the absolute direction and allows us to fit all images to this direction. Some research has focused on animals (Kimchi et al., 2004; Wajnberg et al., 2010; Lohmann et al., 2004) that have a built-in magnetic compass sensor and refer it for navigation.

In the experiment of Figure 2.14, the experimenter changes two parameters. The first is the direction of the magnetic field and the second is the distance to home. The altered magnetic field does not cause a navigation problem in the short trip case and the majority of medium trips, but causes severe problems in the long trip case. the results indicate that the mole uses the self-magnetic information proportionally to the distance of the trip. So, an aligned compass is a useful tool. However, if vision is the only tool available, we have to find the alternatives which only use visual information.

2.6.4.2 Visual Compass

As we do not use idiothetic (internal) information, we need an additional alignment algorithm. In local visual navigation, we can use the visual compass algorithm (Zeil

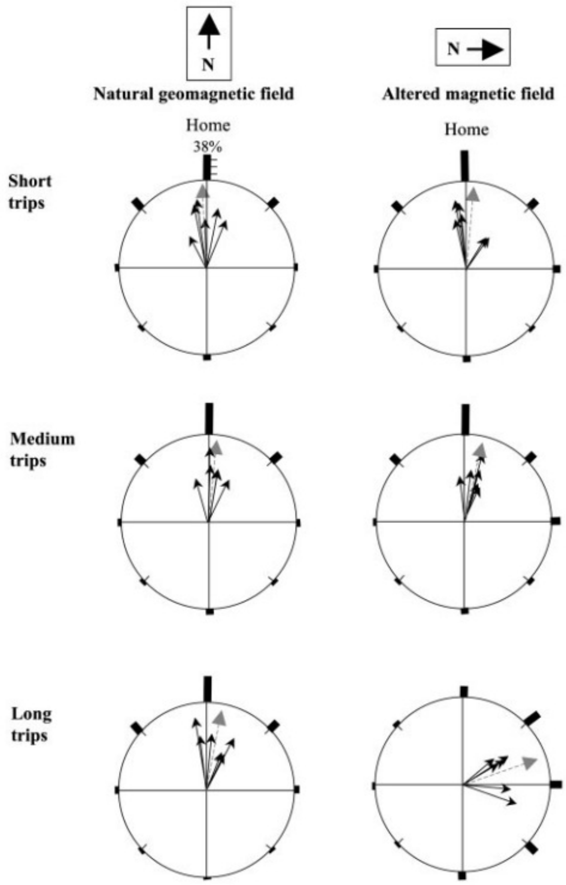


Figure 2.14: Figure shows evidence of the bio compass used by subterranean mammals (reprinted from (Kimchi et al., 2004)). The arrow in the top box shows the direction of the magnetic field (first row: natural, second row: altered). The arrow in the circle shows the movement direction (home at North) of moles in each environment. And the bars around the circle show the number of arrivals as output

et al., 2003) which is based on only two images. The algorithm is very simple. First we obtain one image (home) to use as the norm in direction. Then other images have to be aligned with the first one. Other images are shifted until the minimum errors derived from subtraction between the norm and shifted image are determined. Then the shifted image with minimum error is equal to the aligned output. As we cannot know the magnetic north of the image, we do not need it, because we aligned all of the input images based on the home image. The error is calculated as below.

$$\overline{\theta} = \arg\min_{\theta} \sum_{w=0}^{360^\circ} \sum_{h=1}^{H} |SS(w,h) - CV_{\theta}(w,h)| \tag{2.6}$$

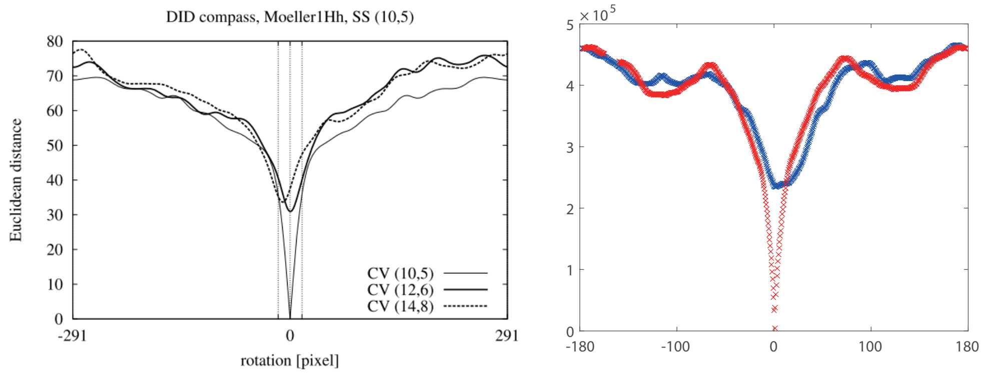


Figure 2.15: The experiment of visual compass. The left figure shows an example of a visual compass (reprinted from (Möller et al., 2007)). Visual compass: For a snapshot (SS) from grid position (10,5) and three different current views (CV), the Euclidean distance under rotations of the snapshot image is shown. All images have image size 583 × 81. Image pixels are in the range [0,1] . Cutoff frequency is 0.05. The right side of the figure shows the example of the visual compass that we test. It counts the errors according to the shifted angle of input images. Blue line shows the case using original image and the minimum case appears in the exact center. Red line uses another image and the aligned angle is almost equal to the blue one. The vertical line shows the minimum error cases, and the intersected point on the x-axis indicates the alignment angle

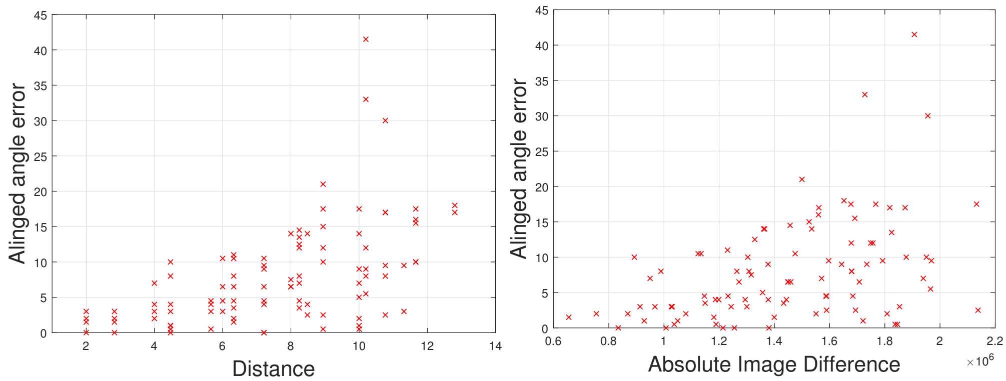


Figure 2.16: Performance of alignment based on the visual compass. First is comparing the output according to the distance between two locations. Second is the effect of absolute image difference.

Figure 2.16 is an example of the visual compass. For the same location (SS), the difference between rotated image and snapshot is zero. If the image is from the current position [CV], there must be difference in information by location and angle, which has the smallest error and is equal to the compass value. An improvement to alignment has been proposed using three-dimensional robot pose changes (Jouir et al., 2015).

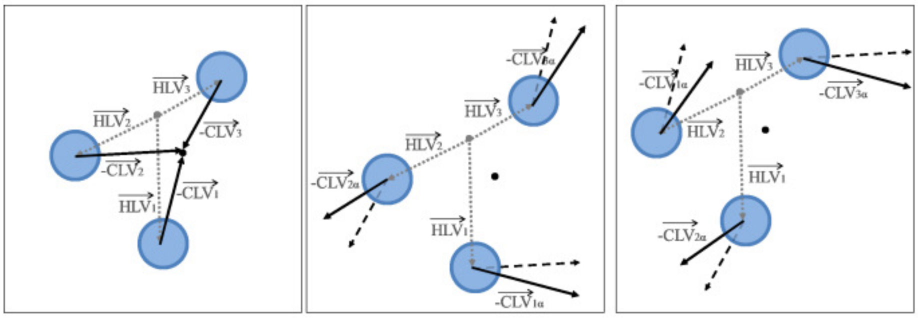
In short, as the image from the vicinity of home site has similar arrangement of landmarks in the image, then we can easily find the compass value so long as we check the difference at all possible angles. But this checking process demands extensive computation time by n times (n is the angle resolution of the visual compass). There are many challenges in to substitute the magnetic compass with a visual compass, and research to solve these challenges using probability models such as the particle filter (Montiel and Davison, 2006), neural network models (el Jundi and Homberg, 2010), etc. Also with the use of a deep auto-encoder to filter out occlusion from dynamic situations has been proposed (Walker et al., 2017).

2.6.4.3 landmark rearrangement

For the method using differences in landmark locations, if the robot has a built-in compass, it finds home by matching between landmarks and estimates the current location to each landmark. Therefore, similar to the visual compass, it finds the estimated locations to each landmark by matching all possible angles. Then we can check the distribution of estimated locations to fixed angle difference and also we can find the case which has the lowest variation of distribution when both sequence and angle differences are fitted. The angle difference in this case will be the compass value. Similar to the visual compass, if both snapshots have enough common factors, then it can find the accurate direction. But the difference is that this method does not use all pixels, but uses extracted landmarks instead.

Figure above shows three cases of different matching. In these cases the current location is estimated by reverse injection using landmark vector matching. Then the agent determines the angle and homing direction when both the sequence and angle are fitted (a). Similar to this method, there are algorithms which find the direction by comparing all possible cases. Therefore landmark matching is used as modification in diverse studies.

We evaluate the problems associated with the compass and possible solutions. First, decide which type of compass to use. If a compass is used, reduce the calculation or if not, use other algorithms (visual compass, landmark arrangement, etc.) to emulate a compass. Use the tool most suited to the purpose and robot.



(a) (b) (c)

Figure 2.17: Figure shows the principle of landmark arrangement matching. The circles indicate the landmarks. Gray dot is home and black dot is estimated current location. Broken arrow shows the ideal direction when the direction is fitted and Solid arrow shows the direction including error.

2.6.4.4 various alignment algorithms

We test various alignment models to compare them to the visual compass. As this part directly affects the overall performance of our model, it is very important to find an optimal alignment algorithm before starting our model. First is the visual compass which is from the research described in (Labrosse, 2006). Based on the linear search, the current view is rotated, one column at a time, and the absolute difference against the reference image is calculated. Then after locating the one having smallest difference, the corresponding angle is selected as the norm of alignment.

Other methods used for testing performance included three items. One (Horst and Möller, 2017) using the computation of the scale-plane stack to treat various sizes of images. In this part, we use the normalized sum of absolute differences (NSAD) and multi-scale. Another (Kuglin and Hines, 1975; Stürzl and Mallot, 2006) uses phase correlation with the row sum method. Since the rotation component causes a horizontal image shift, the phase correlation method which is used for finding time shifts in the time domain can be used. The third method is a modified version using a combination of both phase correlation and scale-plane stack methods.

Figure 2.18 shows the results of the test. The left figure shows the output of each test having various locations and the right figure shows mean values. It is apparent that the visual compass has the best output in the overall region. However, as mentioned previously, we also know that there must be some errors introduced when distant locations

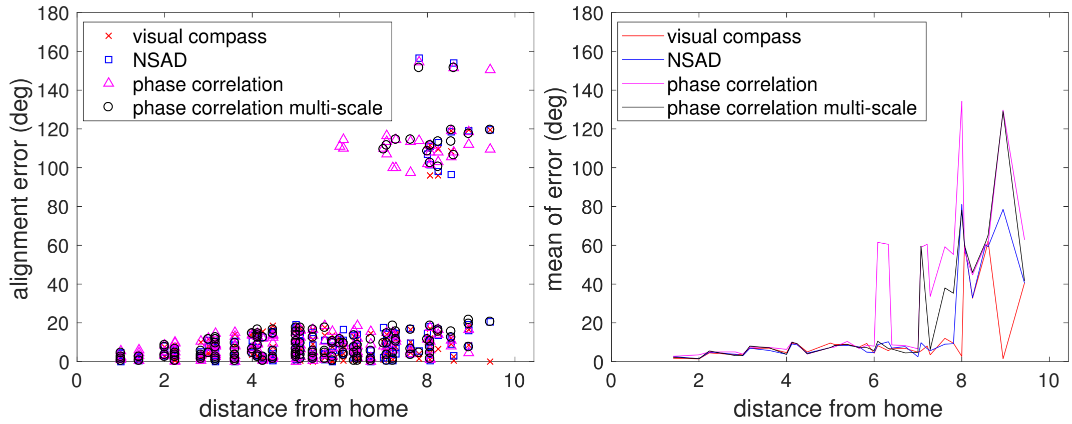


Figure 2.18: Accuracy of four alignment algorithms. The visual compass is marked by red x. NSAD with multi-scale comparison is marked by blue squares. Phase correlation is marked by magenta triangles. The modified version combining both phase correlation with multi-scale is marked by black circles. The left figure shows the alignment error of various locations. Each point has different shape and color for the absolute value of the alignment error between snapshot and current view. The right figure shows the mean of errors. The size of each bin is 2.

are considered.

2.6.5 Problem 5: Landmark Feature Selection

The next problem is the feature selection problem. Features represent visually special information such as landmarks. There are various navigation techniques applied to wheeled vehicles (Widyotriatmo and Hong, 2011) to a wide range of underwater vehicles (Zhang et al., 2016), unmanned aerial vehicles (Zhu et al., 2017a), and spacecraft (Huang and Yan, 2016) along with development of various applicable sensors (Lyshevski, 2017; Delgado et al., 2012). Additional research using various sensors such as vision (Gilg and Schmidt, 1994b), inertia (Wang and Xie, 2015), and RFID (Park and Hashimoto, 2009) has been presented. Therefore selection of features according to the type of sensor is very important.

Based on bio-inspired navigation, we continue to focus on vision. However various visual navigation methods are needed to locate optimal features from vision data. And if the environment is very complex, then we use feature extraction. Then feature extraction can be summarized into 'how and what pixels do we select as features to use in navigation'. Feature extraction as a pre-process of navigation directly affects performance.

2.6.5.1 Object detection

Early versions of visual navigation used some designated object having a special color as a marker (like the example of ALV). Much research has focused on finding the marker or object in an image for visual navigation purposes using object detection (Ess et al., 2010; Lee et al., 2012). However, a challenge is to determine which object to choose and track from an input image. Some solutions use manual selection of a tracking object (CamShift) (Nouar et al., 2006; Wang et al., 2009; Exner et al., 2010). These solutions have been improved with the advent of deep learning to find objects from images based on the Bag-of-Words concept with various structures of neural networks (Chen et al., 2016; Lin et al., 2016; Szegedy et al., 2016; Redmon et al., 2016; Ren et al., 2017) and is directly applied into navigation researches based on SLAM (Zhu et al., 2017b; Gupta et al., 2017b; Dong et al., 2016). However, we think that small insects cannot accurately recognize what objects are included in visual input cues having poor resolution.

2.6.5.2 feature point extraction

Next is the method which is the commonly-used mechanism in feature tracking, SIFT (Lowe, 2004). Methods based on SWIFT try to find a suitable feature instead of an object in the images using changes in intensity. Based on corner characteristics, the point which is selected by feature tracking is always conspicuous regardless of its position. Then we can easily find the same point at different positions, in which is useful for navigation purposes. For example, using a scale invariant feature (SIFT) algorithm which allows the agent to set good landmarks (features) to track (Pons et al., 2007; Ramisa et al., 2011; Luke et al., 2005), improves the quality of feature matching. Saez-Pones et al. suggested an orientation model with the SIFT algorithm applied into a local visual navigation model (Pons et al., 2007). Since it can choose features with robust corners that do not change under various conditions, the selected features are robust to noise and distortions of the image regarded. Figure 2.19 shows an example of finding feature points from an image. The SIFT algorithm not only finds robust feature points, but also connects equal features from two images. However, due to variations

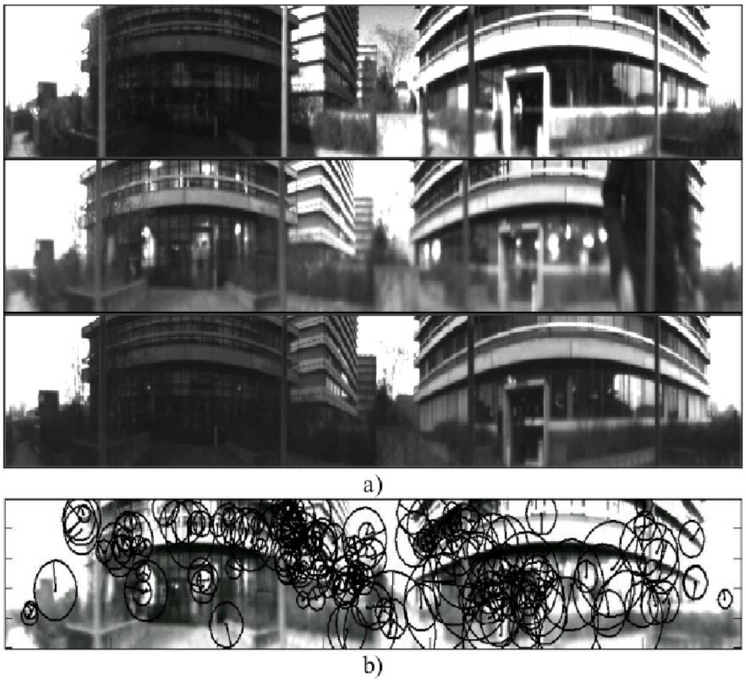


Figure 2.19: Using SIFT algorithm and finding features on an image (Reprinted from (Pons et al., 2007))

in the conservation of features, it cannot guarantee good performance when compared with other methods.

Based on robustness of SIFT, there has been much feature point research, such as HOG (Karmaker et al., 2016), SURF (Bay et al., 2008), BRIEF (Calonder et al., 2012), BRISK (Leutenegger et al., 2011), ORB (Rublee et al., 2011), and variations of SIFT such as PCA-SIFT (Ke and Sukthankar, 2004) and ASIFT (Morel and Yu, 2009). To apply real application, most of them focus on real-time operation fast features (Yang and Cheng, 2012; Rosten and Drummond, 2006) and there are also bio-inspired ones such as the retina mimicking feature called FREAK (Alahi et al., 2012). These are also generally applied into visual navigation (Wu et al., 2016; Murillo et al., 2007; Arroyo et al., 2014) but are not included in holistic methods and not suitable to navigation of animals due to their complexity. Thus, we did not use feature points in this paper.

2.6.5.3 pixel landmark

There are some special cases (Basten and Mallot, 2010) where algorithms need feature extraction but do not use it. In this case, especially, they use all of the pixels along the horizontal line. They consider each pixel as a landmark. In other words, where the

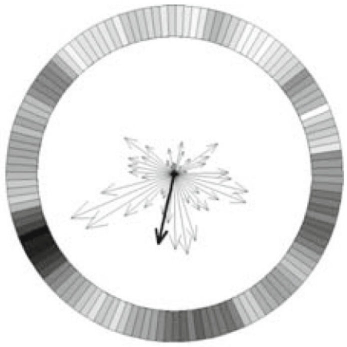
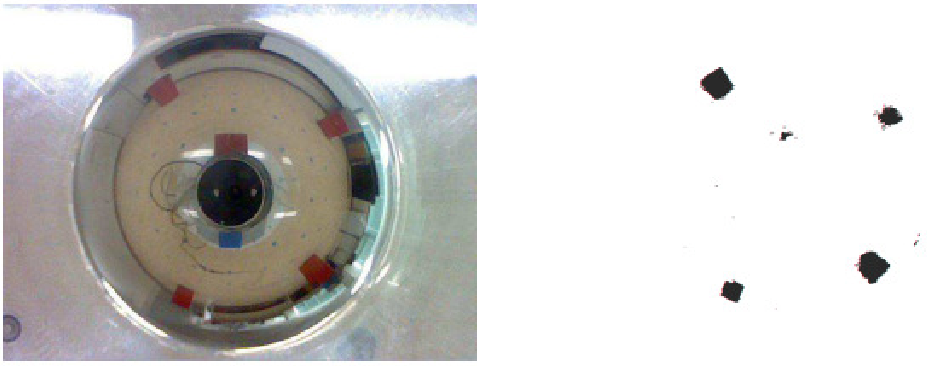


Figure 2.20: Example of the none case (reprinted from (Basten and Mallot, 2010)). The agent creates the landmark vector to all pixels which have the intensity marked by color in the divided ring



(a) (b)

Figure 2.21: Example of finding a landmark using the color value. (a) is the original omnidirectional image and (b) is the output using color value of the landmark's crimson red value. (reprinted from my sector paper)

pixels are from is not important, but the color value in a pixel determines whether it is a landmark.

Color information of pixels can be used in the case where the landmarks have conspicuous characteristics. In other words, if homing navigation of some algorithms needs the location of each landmark, then we create an artificial landmark which has conspicuous color and find it easily in an image by using an RGB value marker. Researchers use this method to prove the possibility of the algorithm because it is simple. However this simplicity limits the range of applications of this method. If the color of a feature is flat or is changed by a luminous source, then the feature cannot be detected. But if we control it properly, it can be used as a minimum unit for visual navigation.

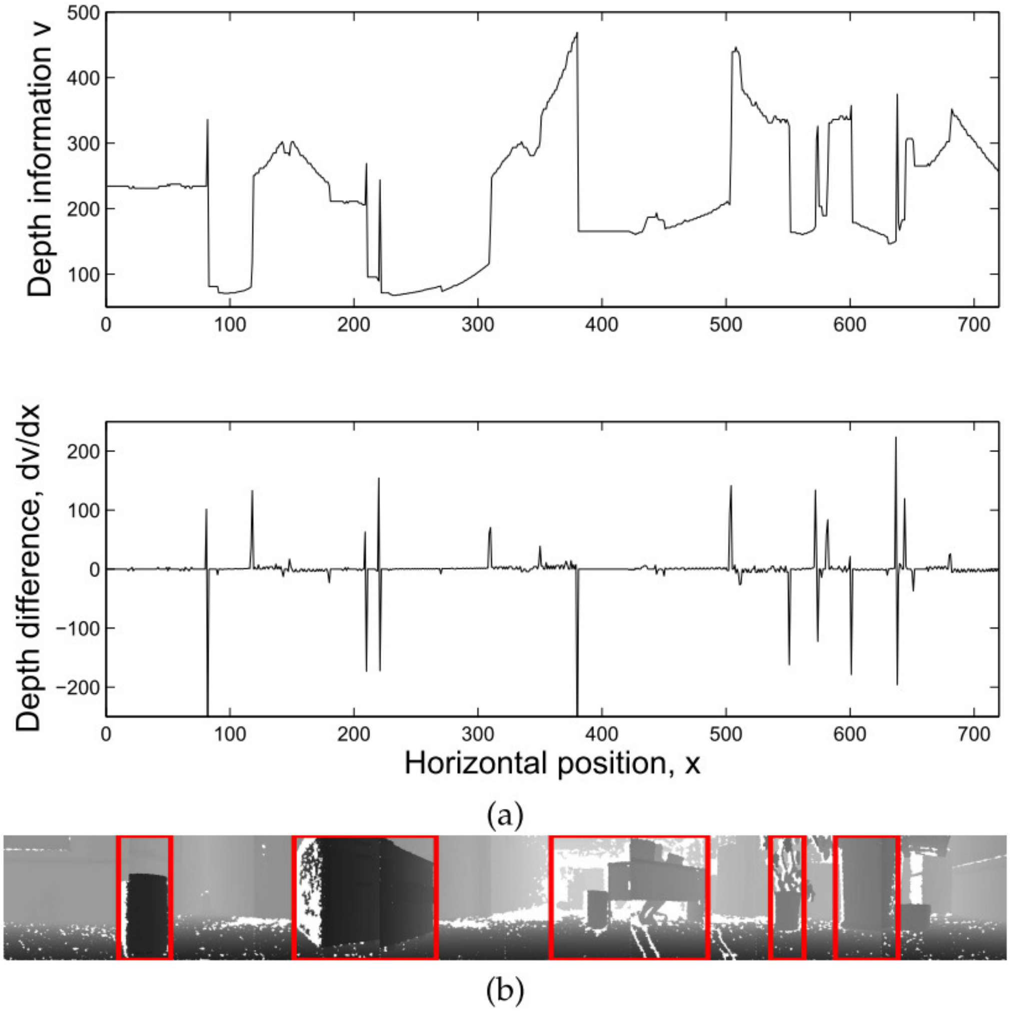


Figure 2.22: Feature extraction using a laser sensor (reprinted from (Yu and Kim, 2010b)). First graph shows the output of the sensor and the second graph shows the first derivation of the output. These outputs are used to find the proper location of a landmark, which is marked in the red box of (b).

2.6.5.4 using distance sensor

One method to determine distance is the using a laser sensor. In this case, the agent uses laser sensor data to determine the landmark location by the difference in distance values. It has low error and directly facilitates finding homing information. However it represents an additional sensor or complex process such as stereo vision. But we think that vision can make range-based navigation robust.

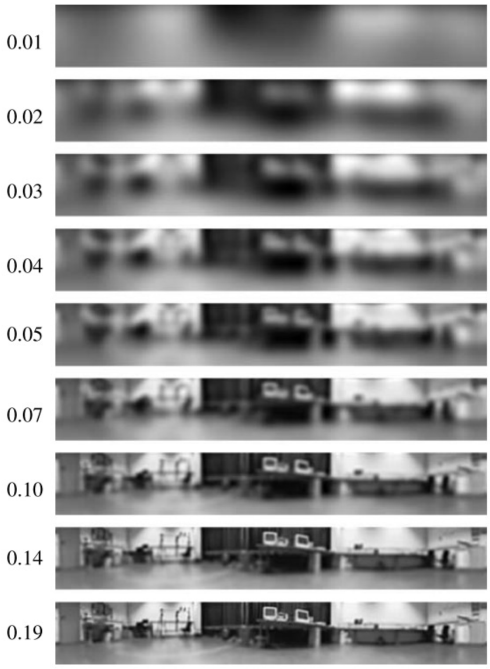


Figure 2.23: Example of Butterworth filtering (reprinted from (Möller et al., 2007)). Effect of varying cut-off frequency of the Butterworth filter

2.6.5.5 filtering

The filtering method is generally used for reducing the complexity of an image. If the environment is very complex, then needless small objects which are significantly changed by every movement interfere with locating useful objects for navigation. Therefore we have to eliminate or reduce these obstacles. Typical examples of tools to accomplish this are filters such as the Gaussian and Butterworth filter (Baddeley et al., 2011; Möller et al., 2007). The repeated application of Gaussian filtering produces a dim image. Then the needless objects are melted into surrounding pixels because they are small and bland. But the useful objects survive and can be used by the agent successfully. Another approach is to use an out of focus camera, because this has a similar effect to the Gaussian filter.

Different to the filters above, there is some research (Nummiaro et al., 2003) into filters which extract the target object and follow it. They use color distribution information and update both estimation and target model to continuously track the object. Small windows including the filter are spread across an image region to locate the target.

A typical example involves Haar-like features introduced for object recognition in vision or face recognition (Viola and Jones, 2001), (Viola and Jones, 2004; Mita et al., 2005; Lienhart and Maydt, 2002). A Haar-like feature is composed of neighboring rectangular regions at a specific location, where the pixel intensities are integrated in each region and the difference between these region sums is calculated. There are several mask patterns depending on the number of rectangular regions or on the distribution of a set of adjacent rectangles. If a part of the target image has a similar pattern to some mask, then the matching score is high. Haar-like features have very simple patterns consisting of a positive or negative sum of pixel intensities. Recent studies show that a collection of those weak mask classifiers can lead to a strong classifier for pattern recognition.

Also, it was reported that the route-following task for a mobile robot can be achieved using Haar-like features (Baddeley et al., 2011; Stürzl and Mallot, 2006). The Haar-like features can visually guide the robot in the right direction to the goal position without a waypoint setup. Similarly, we think the Haar-like features make the image into sum of geometrical figures. The various pre-determined basic figure sets replace the existing complex environment into simple and powerful landmarks. Then we refer to them for easy navigation.

Similar to the mask-based filtering algorithms above, in a biological vision system, there are many receptors in the retina. This vision system has a complex structure, and both cone cells and rod cells are directly connected with an image pixel. Bipolar cells receive synaptic inputs from either rods or cones, or both (Paik and Ringach, 2011), and they are the real basic units for visual perception regarding shape. Many insects demonstrate excellent homing navigation capability even though they have a low-level vision system. We think that even a simple model of visual features might be sufficient for good homing navigation.

Previous research argued that vision occurs in the brain's visual cortex (V1) proposing various models, but recent research indicates differences. Research about the distribution of Bipolar cell structures (Paik and Ringach, 2011) trace the root of these basic image process functions to retinal positions. In this research, they found that the distribution of the receptive field has a noisy hexagonal form (RGC mosaics before LGN) and they suggest the distribution of bipolar cell using a Moire mosaic model without other complex models. Figure 2.24 shows the distribution of RGC dipoles in hexagonal distribution. In this figure, nearest neighbors are connected into dipoles and they operate as line detectors with preferred orientation. The red dot means on-cell and blue dot means off-cell.

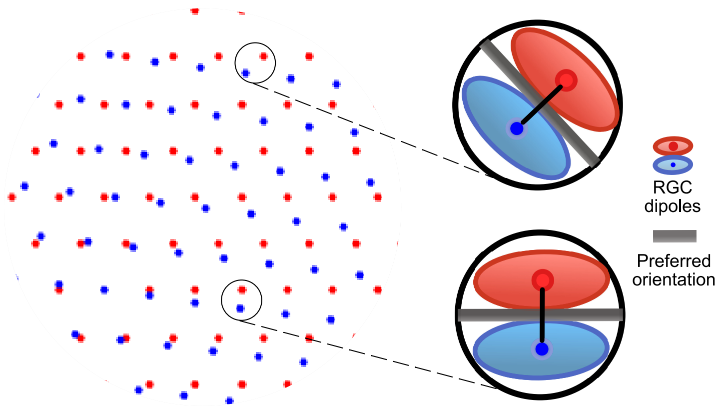


Figure 2.24: Locally, patterns are organized into pairs of dipoles, in which cells of opposite center sign are nearest neighbors of each other. Cortical pooling of inputs from a dipole (relayed by the LGN) would result in simple-cell receptive fields with side-by-side ON and OFF sub-regions. (Paik and Ringach, 2011)

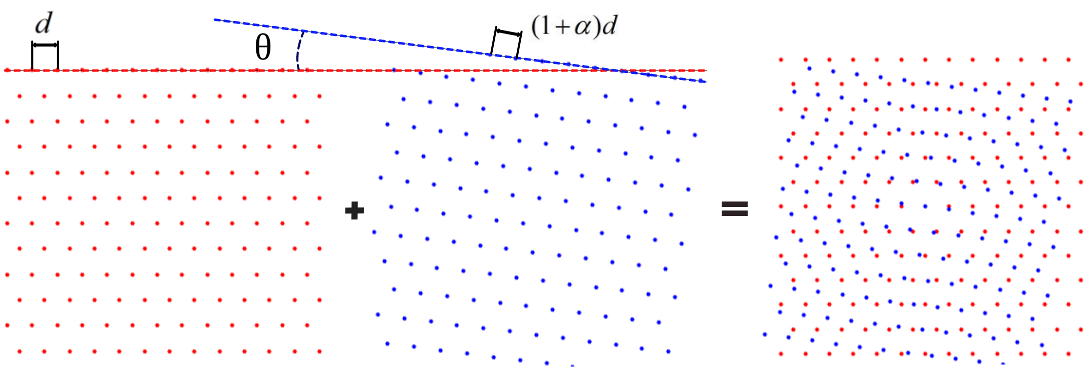


Figure 2.25: Orientation maps as Moiré interference patterns between retinal ganglion cell mosaics. The superposition of two hexagonal lattices results in a periodic interference pattern. (Paik and Ringach, 2011)

If there are specific distributions of bipolar cells in the retina, then the above output can easily be achieved. Figure 2.25 shows the principle of orientation mapping using a Moire mosaic. Similarly, there are two types of dots and there are two differences between two distributions. One is the angle difference between two patterns marked as θ. The other is the distance between neighboring cells. These two factors create a combination that has a hexagonal pattern. From this argument, the brain uses basically line-detected information from the sum of the bipolar cell distributions that are basically installed in the retina. We think that we can use this receptor system by using organized bipolar-like mask generation.

2.7 The importance of these researches

In this chapter, we surveyed about the various histories of bio-inspired visual navigation. Compared with conventional SLAM techniques, bio-inspired snapshot-based models have simplicity but not enough robustness. Those researches successfully reduce the complexity but have various problems that we introduced above parts. However, we have question about the completeness of current models with these problems.

These are Localization, Occlusion problem, observing FOV, Alignment and proper Feature selection. However, we think that geographic map of the localization problem can be replaced by topological maps including multiple snapshot images based on the origin of local visual navigation. Therefore, if we can divide the overall area into small sections successfully and our models have robustness in each section, global performance can be more reliable.

Also, in this field, alignment problems are solved by alignment algorithms that we said. Typically, a visual compass shows robustness and also is partially applied into warping model. We also select the visual compass in alignment by comparing various models and applying it into our models.

However, there are three problems, the first one being the 'feature selection' problem. Obviously, there has been a lot of research using various sensor data, but we cannot find a simple model combining range data with visual information in a snapshot model. We think that if the insect uses both range and vision then it also has simple form. Therefore, based on the snapshot model, we apply moment function from physics combining two measurements without increasing complexities in chapter 3.

The second of the three problems, is the FOV (field of view) problem. Basically, a lot of the models using visual information about movement in a horizontal direction are able to find the proper size for the FOV, thus showing the best output. On the other hand, various ants have almost 1000 ommatidia (Australian desert ant called Melophorus Bagoti : 840-1180 (Schwarz et al., 2011) and the desert ant called Cataglyphis : 940 (Zollikofer et al., 1995)) with fixed size. Therefore, we designed vision receptors like these 2 types of ant, and tried to apply them by using the outputs as feature into the homing problem in chapter 4.

Finally, the third is the occlusion problem. Based on the simplicity of snapshot model, we cannot find a perfect method for this problem without feature matching. Therefore, we tried to find the robustness in occluded situation and the basic principle of the snapshot-based homing, without clustering, so then it can be directly applied into filtering occlusion from observation. We interpret each pixel as a single pixel-wise landmark and find mathematically convexity. Also, we tested the effect of matching by using a simple sequence matching algorithm called DTW in chapter 5.

According to reasons above, we designed new models in each chapter to find answers for each question and succeeded in finding new robust models. Therefore, in this dissertation, we need state-of-art techniques to check robustness. These is the DID (Descent in Image Distance) method (Zeil et al., 2003; Möller and Vardy, 2006), MinWarping algorithm (Horst and Möller, 2017; Fleer and Möller, 2017), COMALV (Center-Of-Mass ALV) (Mangan and Webb, 2009; Hafner, 2001) and MCOMALV (Modified Center-Of-Mass ALV).

DID : In this paper, we use 720x100 pixels with both 0.5 angular resolution in horizontal direction and $\pm$50 pixels around horizontal line in vertical direction.

MinWarping : In this paper, we use 720x100 pixels with both 0.5 angular resolution in horizontal direction and $\pm$50 pixels around horizontal line in vertical direction.

COMALV : In this paper, we convert image including 720x100 pixels into 90 x 80. The it has both 4 degree angular resolution and $\pm$40 pixels around horizontal line in vertical direction.

MCOMALV : Combining with DID approach, we modified the original COMALV into MCOMALV (Modified-COMALV) by using three reference images. Similar with DID, we extracted 90 ommatidia output from three images and calculated each of the differences with the current one. Then, homing direction is calculated by finding the ratio between the three differences. In this paper, we use the same setting in input visual cue with COMALV.

2.8 Summary of Chapter 2

In this chapter, we examined various research studies regarding visual navigation. Based on the snapshot model, these algorithms have both robustness and simplicity that can be easily applied into navigation. There have been many different models in local homing navigation according to each characteristic and they can be classified into different categories, as previously discussed in this chapter. To present our new models, we evaluated overall navigation algorithms and found common problems. Our goal is to develop a new navigation algorithm imitating insect navigation that is both robust and simple. The proposed model is designed to partially solve the common problems observed in state of the art methods.

Chapter 3

Visual Navigation Using Moment Models

In this chapter, we apply a concept of moment function to our new navigation model based on snapshot model of holistic method. Through this part, we aim for checking the effect of distance factor in navigation problem. For example, one of the research of SLAM (Kwon and Song, 2011b) uses elevation information to invent moment model, called Elevation Moment of Inertia (EMOI) to reduce the data size. It reveals that the EMOI can be the representative value at each position which can reduce the surrounding range value and height information into one scalar EMOI value. Then, first, we expand this EMOI concept (Lee and Kim, 2016b) to build moment model for homing navigation by using various feature values. We use both color with range measurement for checking the effect of distance factor in navigation problem. Second, we made new models with various forms by making generalized version of the first model. We make multiple moment models with mathematical proof of our overall models. Third, based on these analyzed moment models, we check the convergence patterns in both simulation and robot experiments. Then we add appendix to explain in details. The one of these researches is published (Lee and Kim, 2017a) and another one is under the preparation for submission(Lee and Kim, 2017b).

3.1 Moment Model for Homing Navigation

In physics, the area moment of inertia is a property of an area which reflects how its points are distributed. By analogy, the area moment is defined as a distribution of point measurements in our navigation model. For a given set of landmarks in the environment, we analyze the landmark distribution as a combination of their positions and features. The color intensity or height of landmarks can be feature candidates.

The color of visual cue is the feature used in this paper. We define the moment measure M as follows:

$$M = \sum_{i=1}^{N} {r_i}^2 C_i = \sum_{i=1}^{N} ((x-a_i)^2 + (y-b_i)^2) C_i \tag{3.1}$$

where there are N landmarks, r_i is the range value of the i-th landmark, that is, the distance from the current location (x,y) to the landmark location (a_i, b_i), and C_i is the feature value, for example, color intensity of the i-th landmark.

The above measure is similar to the area moment of inertia in physics, $\int r^2 dm$. We can also see this measure as a potential function built with a set of landmarks. From that, we can find the gradient as the first derivative of potential function as follows:

$$\nabla M = (\frac{dM}{dx}, \frac{dM}{dy}) = (M_x, M_y) = \sum_{i=1}^{N} (2(x-a_i)C_i, \quad 2(y-b_i)C_i) \tag{3.2}$$

where this gradient vector indicates the change of potential function corresponding to the current position (x,y). To find the minimum convergence point with the gradient, we calculate the determinant of Jacobian matrix

$$J(\nabla M) = \begin{pmatrix} \frac{d^2M}{dx^2} & \frac{d^2M}{dxdy} \\ \frac{d^2M}{dxdy} & \frac{d^2M}{dy^2} \end{pmatrix} = \begin{pmatrix} M_{xx} & M_{xy} \\ M_{yx} & M_{yy} \end{pmatrix} = \sum_{i=1}^{N} \begin{pmatrix} 2C_i & 0 \\ 0 & 2C_i \end{pmatrix} \tag{3.3}$$

where this Jacobian matrix is produced from the second-order differential of the gradient equation. M_{xx} is a second-order differential with respect to x and M_{yy} with respect to y. M_{xy} is equal to M_{yx} and they are zero.

The determinant of Jacobian matrix is calculated as

$$\det(J) = (\sum_{i=1}^{N} 2C_i) \times (\sum_{i=1}^{N} 2C_i) = (\sum_{i=1}^{N} 2C_i)^2 > 0 \tag{3.4}$$

We assume that each feature value (C_i) is positive. The sign of the 2nd derivatives of potential function is positive as shown below:

$$\mathrm{M}_{xx} = \mathrm{M}_{yy} = \sum_{i=1}^{N} 2C_i > 0 \tag{3.5}$$

From the above property, there is only one global convergence (minimum potential) point with its gradient zero, and the determinant of Jacobian is positive. Let (X,Y) be the convergence point. Then

$$\nabla M(X,Y) = \sum_{i=1}^{N} (2(X-a_i)C_i,\ 2(Y-b_i)C_i) = 0 \tag{3.6}$$

The position (X,Y) is calculated as

$$X = \sum_{i=1}^{N} a_i C_i / \sum_{i=1}^{N} C_i, \quad Y = \sum_{i=1}^{N} b_i C_i / \sum_{i=1}^{N} C_i \tag{3.7}$$

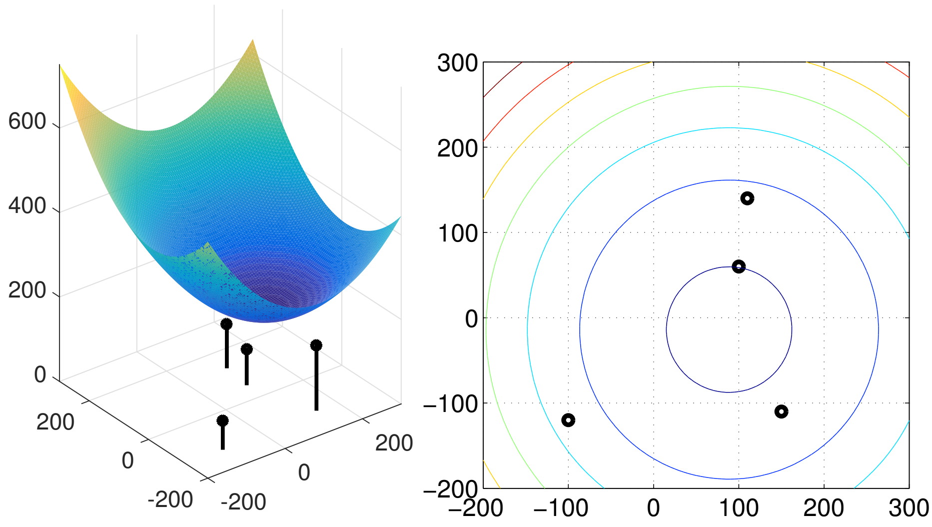
where the convergence point (X,Y) is the weighted average of landmark positions with respect to the landmark features, that is, the center of the landmark distribution. The moment measure based on features of landmarks has unique convergence point (X,Y), regardless of any current position (x,y).

Thus, we argue that if there is no environmental change or no occlusion observed as the robot moves, then we can find the same convergence point in spite of any movement or any change in angular position. To guarantee the unique convergence point, the feature value should be positive. In our experiments, the landmark characteristics are defined as the height of landmarks or color intensity, which is positive. The moment measure is an index of landmark distribution and its center of distribution can easily be estimated as an invariant feature, which will be useful in homing navigation. In Figure 3.1, the surface of potential function is convex-shaped, and the unique convergence point is available. Various types of feature (C_i) values are available and the convergence point can change depending on the features.

3.1.1 Homing Vector Using Moment Model

We introduce how to estimate the homing vector using the moment function. We assume there is a reference compass available. Each landmark has the feature value and range information. An agent can observe a distribution of landmarks at a given position. Assume the same landmarks are observed at any position in the environment. We take the above global convergence point as the reference point to estimate the homing vector.

If there are N landmarks observed at the current position $P = (x,y)$, their relative distance $r_i = ||(a_i - x, b_i - y)||$ and the feature value C_i are measured for $i = 1,...,N$, where



(a) (b)

Figure 3.1: Moment measure as a potential function where the landmark height is regarded as a feature value (left: potential function, right: contour line

$(a_i - x, b_i - y)$ is the estimated landmark position in the coordinate with origin at the current position. The reference point vector $R = (X_c, Y_c)$ can be calculated by Equation (3.7). In a similar way, N landmarks are observed at the home position $\bar{H} = (H_x, H_y)$. Their relative distance $r'_i = ||(a'_i - H_x, b'_i - H_y)||$ and the feature value C'_i are measured for $i = 1, ..., N$, where $(a'_i - H_x, b'_i - H_y)$ is the estimated landmark position with origin at the home position $\bar{H}$. The reference point vector $R' = (X_h, Y_h)$ can be calculated by Equation (3.7) again at the home position.

Then, we find the relation for homing vector $\overrightarrow{H}$:

$$\overrightarrow{H} = \bar{H} - P \simeq R - R', \tag{3.8}$$

since we assume that the same reference point is estimated irrespective of any observation point, that is the two vectors R and R' should end at the same reference point, starting from the different positions, the current position and the home location (a little deviation of the reference points may be observed by noisy sensor readings or landmark occlusions).

At an arbitrary position $P = (x, y)$, a mobile robot has information of the relative distance and the visual features with a laser sensor and a vision camera. Equation (3.7) has absolute coordinate representation, and so, we evaluate the convergence point in the coordinate with origin at the observation point.

$$X_c = \sum_{i=1}^{N} (a_i - x)C_i / \sum_{i=1}^{N} C_i, \quad Y_c = \sum_{i=1}^{N} (b_i - y)C_i / \sum_{i=1}^{N} C_i \tag{3.9}$$

where $P = (x,y)$ is the current observation point and $(a_i - x, b_i - y)$ is the relative distance of the i-th landmark in the current view. Similarly, the convergence point can be evaluated in the home coordinate as follows:

$$X_h = \sum_{i=1}^{N} (a'_i - H_x)C'_i / \sum_{i=1}^{N} C'_i, \quad Y_h = \sum_{i=1}^{N} (b'_i - H_y)C'_i / \sum_{i=1}^{N} C'_i \tag{3.10}$$

where (H_x, H_y) is the home location. Then, the difference of the two reference points measured at two observation points (the home location and the current position) is given by:

$$\begin{aligned}
R - R' &= (X_c, Y_c) - (X_h, Y_h) && (3.11)\\
&= \left(\frac{\sum (a_i - x)C_i}{\sum C_i}, \frac{\sum (a_i - y)C_i}{\sum C_i}\right) - \left(\frac{\sum (a'_i - H_x)C'_i}{\sum C'_i}, \frac{\sum (a'_i - H_y)C'_i}{\sum C'_i}\right) && (3.12)\\
&\simeq \left(\frac{\sum (a_i - x)C_i}{\sum C_i}, \frac{\sum (a_i - y)C_i}{\sum C_i}\right) - \left(\frac{\sum (a_i - H_x)C_i}{\sum C_i}, \frac{\sum (a_i - H_y)C_i}{\sum C_i}\right) && (3.13)\\
&= \left(\frac{\sum (H_x - x)C_i}{\sum C_i}, \frac{\sum H_y - y)C_i}{\sum C_i}\right) && (3.14)\\
&= (H_x - x, H_y - y) = \vec{H} - P && (3.15)
\end{aligned}$$

where it is assumed that the same landmarks and the same visual features are observed at any position, $(a_i, b_i) = (a'_i, b'_i)$, $C_i = C'_i$ for $i = 1, ..., N$. Hence, the homing vector $\overrightarrow{H}$ can be estimated by the above property,

$$\overrightarrow{H} = (X_c, Y_c) - (X_h, Y_h) \tag{3.16}$$

Each position defines its own reference map, but there exists a unique convergence point that is the same position regardless of any coordinate. By the convexity of the moment potential function, the minimal potential point can be reached from any position. We provided a proof that the homing vector calculated by the above model can reach the home position from any position, if the environment is isotropic, that is all landmarks and their features are invariantly observed at any position.

3.1.2 Moment Model with Multiple Features

If there are multiple features available for landmarks, then we can build a separate moment model for each feature. The set of moment models will lead to independent reference points, but we can assume that the distribution of each feature in the environment will be almost equal for any measured position if the environment is isotropic, that is the majority of landmarks are commonly observed in the environment. The homing vector for each feature can be voted together, which can help estimate homing direction more accurately.

We can test the moment model with RGB color intensities, three visual features for each pixel. The image colors provide three different features, red, green and blue color intensity for each pixel. The landmark feature C_i can thus have three components. The moment measure for each feature, red, blue and green intensity, respectively, is defined as follows:

$$M_R = \sum_{i=1}^{N} {r_i}^2 R_i = \sum_{i=1}^{N} ((x-a_i)^2 + (y-b_i)^2) R_i \tag{3.17}$$

$$M_G = \sum_{i=1}^{N} {r_i}^2 G_i = \sum_{i=1}^{N} ((x-a_i)^2 + (y-b_i)^2) G_i \tag{3.18}$$

$$M_B = \sum_{i=1}^{N} {r_i}^2 B_i = \sum_{i=1}^{N} ((x-a_i)^2 + (y-b_i)^2) B_i \tag{3.19}$$

where (a_i, b_i) for $i = 1, ..., N$ are the landmark position with respect to the current position $P = (x, y)$ and (R_i, B_i, G_i) are the color intensity for the i-th landmark.

Then, the above three measures lead to three reference points at a given position $P = (x, y)$, using Equation (3.9).

$$(\mathrm{X}_R, Y_R) = (\frac{\sum (a_i - x) R_i}{\sum R_i}, \frac{\sum (b_i - y) R_i}{\sum R_i}), \tag{3.20}$$

$$(\mathrm{X}_G, Y_G) = (\frac{\sum (a_i - x) G_i}{\sum G_i}, \frac{\sum (b_i - y) G_i}{\sum G_i}), \tag{3.21}$$

$$(\mathrm{X}_B, Y_B) = (\frac{\sum (a_i - x) B_i}{\sum B_i}, \frac{\sum (b_i - y) B_i}{\sum B_i}), \tag{3.22}$$

Three reference points can be determined both at the current position and at the home location. The difference of the reference points can estimate the homing direction.

The home vector $\overrightarrow{H}$ via the three reference points using the color intensities can be derived as a combinational form,

$$\overrightarrow{H} = (\overrightarrow{H_R} + \overrightarrow{H_G} + \overrightarrow{H_B})/3 \tag{3.23}$$

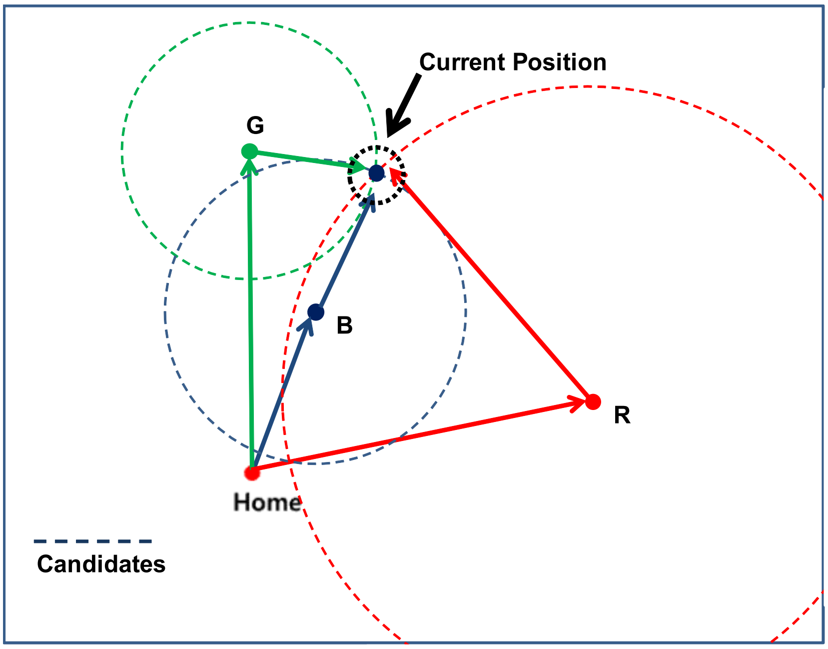


Figure 3.2: Figure shows the principles of localization based on distances to convergence points in reference map. R is the convergence point with using red information. B and G are using blue and green. Dotted lines show the candidates from each convergence point with distance from current point to convergence point

where $\overrightarrow{H_R}$ is calculated with red color intensity, $\overrightarrow{H_G}$ with green and $\overrightarrow{H_B}$ with blue, using Equation (3.16).

As shown above, the color intensity of pixels can be applied to the moment model with multiple features. We can extend the moment measure into that with various visual features. The visual feature C_i allows any characteristics of landmarks, and also, multiple features can derive multiple homing vectors. The sum of the homing vector for each feature can be effective on noisy feature readings. RGB color space can be converted into another space, for example HSV space, and each feature can make separate homing vectors. Furthermore, to handle noisy sensor readings, we can allow a cut-off threshold for a feature value, and some feature can be set to $C_i = 0$. This has the effect of choosing a set of landmarks in the omnidirectional view, instead of using the whole pixels. If $C_i = 0$ or $C_i = 1$ with the range value $r_i = 1$, the moment model becomes similar to the ALV model (Lambrinos et al., 2000). If $C_i = 0$ or $C_i = 1$ with continuous range value $r_i > 0$, the model is similar to the DELV model (Yu and Kim, 2012).

3.1.3 Navigation Using Moment Function with Multi-Dimension Form

3.1.3.1 Area integration Moment navigation

Our prior model treats each measurement as a particle having feature value instead of mass. Like a star in the space, it can find the convergence point and homing direction. Attached to that, a material called 'Dark matter' that we cannot directly observe composes about 80 percent of the mass of the universe. Attached with this concept of black matter, We expand our moment idea. In our past model, omni directional measurement is used as simple dot having feature. Then this area, where we try to finding homing direction, can be a closed space with boundaries. We fill this empty space with virtual landmarks having equal feature value to the one of end point in each direction. Although these imaginary features are not arranged in equal density, but we can approximate some effects of filling the space. We call it as simple area integration moment navigation, following this assumption, it can be calculated by using higher dimension term ($n = 3$) in moment function. Therefore we check the convexity of multi-dimension cases of moment model in next section.

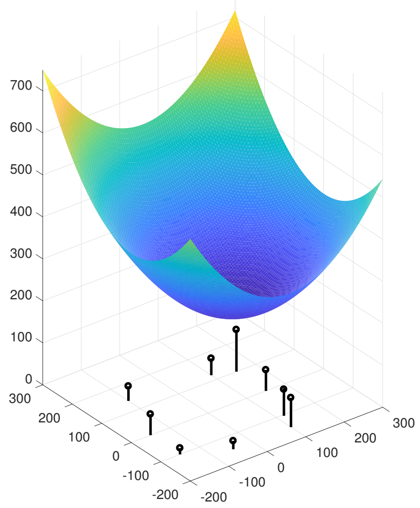
3.1.3.2 Convexity of Multidimensional moment function

In this part, we prove the condition about convexity of various moment function using both radius and characteristic value. We design general type of moment function about changing dimension number like

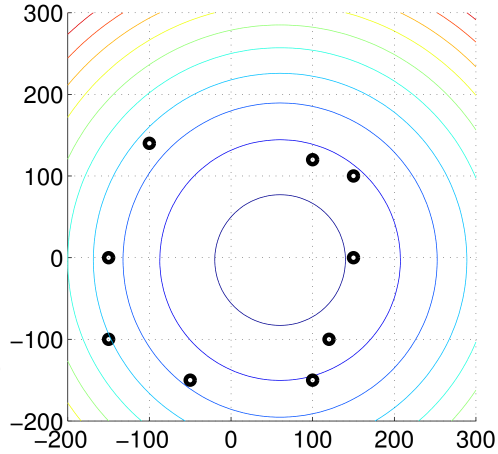
$$M_n = \sum_{i=1}^{N} M_{i,n} = \sum_{i=1}^{N} r_i^n C_i = \sum_{i=1}^{N} ((a_i - x)^2 + (b_i - y)^2)^{\frac{n}{2}} C_i \tag{3.24}$$

where n is dimension number and multiplier to range value (r) and M_n is moment function with this dimension. N is the number of measurements, C_i for feature value of i-th measurement, (a_i, b_i) for location of measured feature and (x, y) for current location of robot. Then we solve it to find global convergences in this condition according to changing dimension number. Then the output is

$$f \left[\begin{array}{cccc} n \geq 1 & \det(J(\nabla M_K(X))) > 0 & M_{K,XX}(X) > 0 & success \\ 0 < n < 1 & unknown & M_{K,XX}(X) < 0 & unknown \\ n \leq 0 & unknown & M_{K,XX}(X) < 0 & fail \end{array} \right. \tag{3.25}$$



(a)



(b)

Figure 3.3: The simulation of unique convergence example on the field. Height value is used as feature of each landmark. First one shows the surf output and the second shows contour line.

where three conditions having outputs of both determinant of Jacobian and quadratic differential about X ($M_{k,XX}$). In this part, we can check that first case ($n \geq 1$) has convexity and the others are not. Because, those two conditions are requirements for unique global minimum. Therefore we can say that the convexity of moment functions can be realized when positive C_i and dimension number over one. Figure 3.3 shows simple simulation about unique convergence point when we use $n = 2$. There are some objects having height and we use it as feature value. Overall distribution of potential field has unique convergence around (80,0). (You can check the detail procedures of above calculation in appendix.)

3.1.3.3 Generalized version of homing navigation

Our prior model is about the navigation using moment function with second order ($n = 2$) case that can find both homing direction and current location algebraically. Similar to the case of convexity proof about generalized version of moment function, we try to make generalized version of homing navigation with various moment function. Then general moment potential with n dimension,

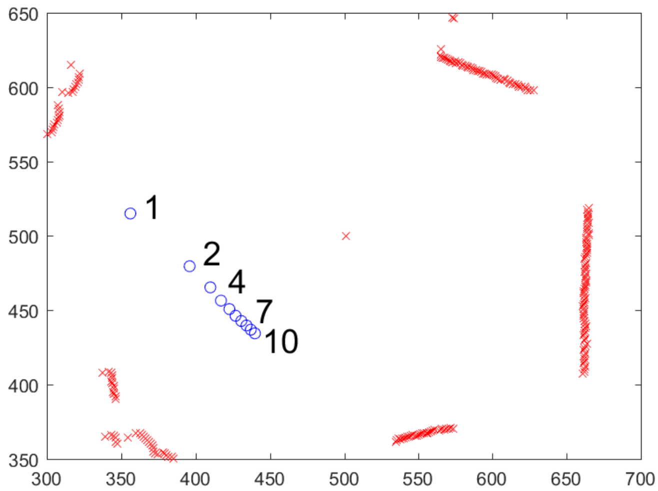


Figure 3.4: Experiment about the location of convergence point according to the changes of order number (n). The environment of experiment is equal to the ones of figure 11 and 12. Red dots are the minimum points with text figuring order number.

$$M_n = \sum_{i=1}^{N} (r_i)^n C_i \tag{3.26}$$

To solve this equation, we change the variable $k = n/2$ like

$$M_k = \sum_{i=1}^{N} (r_i{}^2)^k C_i = \sum_{i=1}^{N} \left((a_i - x)^2 + (b_i - y)^2\right)^k C_i \tag{3.27}$$

Next, similar with prior courses, we calculate derivatives of this potential field like,

$$\nabla M_k = \sum_{i=1}^{N} k\{(a_i - x)^2 + (b_i - y)^2\}^{k-1} [2(a_i - x)C_i, 2(b_i - y)C_i] \tag{3.28}$$

When this dimension number is equal to 2, we can find where the convergence point $((X_{c,n}, Y_{c,n}))$ is. However, when this value is over 2, it cannot be calculated. But, there must be a unique minimum point and then it satisfies

$$\begin{aligned} \sum_{i=1}^{N} k\{(a_i - X_{c,n})^2 + (b_i - Y_{c,n})^2\}^{k-1} \times 2(a_i - X_{c,n})C_i = 0 \\ \sum_{i=1}^{N} k\{(a_i - X_{c,n})^2 + (b_i - Y_{c,n})^2\}^{k-1} \times 2(b_i - Y_{c,n})C_i = 0 \end{aligned} \tag{3.29}$$

Therefore we can check that potential curve has global minimum at $(X_{c,n}, Y_{c,n})$ but cannot calculate the location of this point at once. We do some additional experiment about the location of convergence point according to the changes of order number (n). As calculation about the minimum point of moment function can be possible only when $n = 2$, we find the minimum points in other cases by area search. Figure 3.4 shows the movement of minimum point.The output is not linear. Then we apply numerical search using gradient descent model like

$$\begin{aligned} (X_{t+1}, Y_{t+1}) &= (X_t, Y_t) - \gamma_t \nabla M_k((X_t, Y_t)) & (3.30) \\ \gamma_t &= \frac{(X_t - X_{t-1}, Y_t - Y_{t-1})^T [\nabla M_k(X_t, Y_t) - \nabla M_k(X_{t-1}, Y_{t-1})]}{||\nabla M_k(X_t, Y_t) - \nabla M_k(X_{t-1}, Y_{t-1})||^2} & (3.31) \end{aligned}$$

where (X_t, Y_t) is updated location of anticipated convergence point in step t and γ is learning rate in gradient descent. It is terminated when the amount of change in each step is decreased under 0.1. After searching, we use it as convergence point and it is unique and fixed location like

$$R - R' = (X_c, Y_c) - (X_h, Y_h) = (H_x - x, H_y - y) = \bar{H} - P \tag{3.32}$$

where it is assumed that the same landmarks and the same visual features are observed at any position, $(a_i, b_i) = (a'_i, b'_i)$, $C_i = C'_i$ for $i = 1, ..., N$. Hence, the homing vector $\overrightarrow{H}$ can be estimated by the above property,

$$\overrightarrow{H} = (X_c, Y_c) - (X_h, Y_h) \tag{3.33}$$

Also, we try to reduce this process calculating the location of convergence point into derivative form without repetition. In this part, without precise location of convergence point, it calculates homing direction by following gradient of each point like

$$\overrightarrow{H_k}(x, y) = -\nabla \overline{M}_k = \nabla M_k(x, y) - \nabla M_k(x_{home}, y_{home}) \tag{3.34}$$

where $\overrightarrow{H_k}(x, y)$ is calculated homing vector at location (x, y) with dimension ($k = n/2$). Then this form is designed to align convex moment curve fitted to home position $(X_0(x_{home}, y_{home}))$. Also, this type of moment function satisfies strictly convex condition at home position therefore we can use it in homing. Because it satisfies three convex conditions which are

$$\det(J(\nabla \overline{M}_k(X))) > 0 \quad \overline{M}_{k,xx}(X) > 0 \quad \nabla \overline{M}_k(X_0) = 0 \tag{3.35}$$

where

$$J(\nabla \overline{M}_k) = \begin{pmatrix} \frac{d^2 \overline{M}}{dx^2} & \frac{d^2 \overline{M}}{dxdy} \\ \frac{d^2 \overline{M}}{dxdy} & \frac{d^2 \overline{M}}{dy^2} \end{pmatrix} = \begin{bmatrix} \overline{M}_{k,xx} & \overline{M}_{k,xy} \\ \overline{M}_{k,yx} & \overline{M}_{k,yy} \end{bmatrix} \tag{3.36}$$

this Jacobian matrix can be calculated by quadratic differential of each term including $\overline{M}_{xx}$ for quadratic differential about x and $\overline{M}_{yy}$ about y. Then our model can satisfy those three convergence conditions to home position, the homing vector ($\overrightarrow{H}$) is calculated by subtraction between those two vectors. If there is no occlusion and measurement error, it can calculate proper homing direction. Also, different to the second order case, the other cases cannot find current location. But the overall flow of this potential field can reach to the home position by convexity. We show detail procedure in appendix.

3.1.3.4 Generalized area integration model having uniformity

Through the above parts, we can find the convergences of moment function having various dimension numbers. In this part, we modify the simple area integration model to having equal density and analyse the effects of filling operation in homing navigation. To make equal density, we modify the prior simple model into integral form like

$$M_f = \int_0^{2\pi} \int_0^{r_\theta} r^2 C_\theta \rho dA d\theta = \int_0^{2\pi} C_\theta \int_0^{r_\theta} r^2 \rho (r dr d\theta) \tag{3.37}$$

where the M_f is area integration moment level, N for the number of landmarks, C_i for feature value of i-th measurement, r_i for distance to feature and ρ, ρ' for normalize constant. Then it can be arranged into

$$M_f = \rho \int_0^{2\pi} C_\theta \int_0^{r_\theta} r^3 dr d\theta = \rho \int_0^{2\pi} \frac{1}{4} r_\theta^4 C_\theta d\theta \cong \rho' \sum_{i=1}^{N} r_i^4 C_i \tag{3.38}$$

Then we can observe the final output having more higher dimension number ($n = 4$) with normalization constant. Next, we attach the multidimensional model into this output like

$$M_{f,n} = \int_0^{2\pi} \int_0^{r_\theta} r^n C_\theta \rho dA d\theta = \int_0^{2\pi} C_\theta \int_0^{r_\theta} r^n \rho (r dr d\theta) \tag{3.39}$$

where the $M_{f,n}$ is area integration moment level, N for the number of landmarks, n for dimension number, C_i for feature value of i-th measurement, r_i for distance to feature and ρ, ρ' for normalize constant.

$$M_{f,n} = \rho \int_0^{2\pi} C_\theta \int_0^{r_\theta} r^{n+1} dr d\theta = \rho \int_0^{2\pi} \frac{1}{n+2} r_\theta^{n+2} C_\theta d\theta \tag{3.40}$$

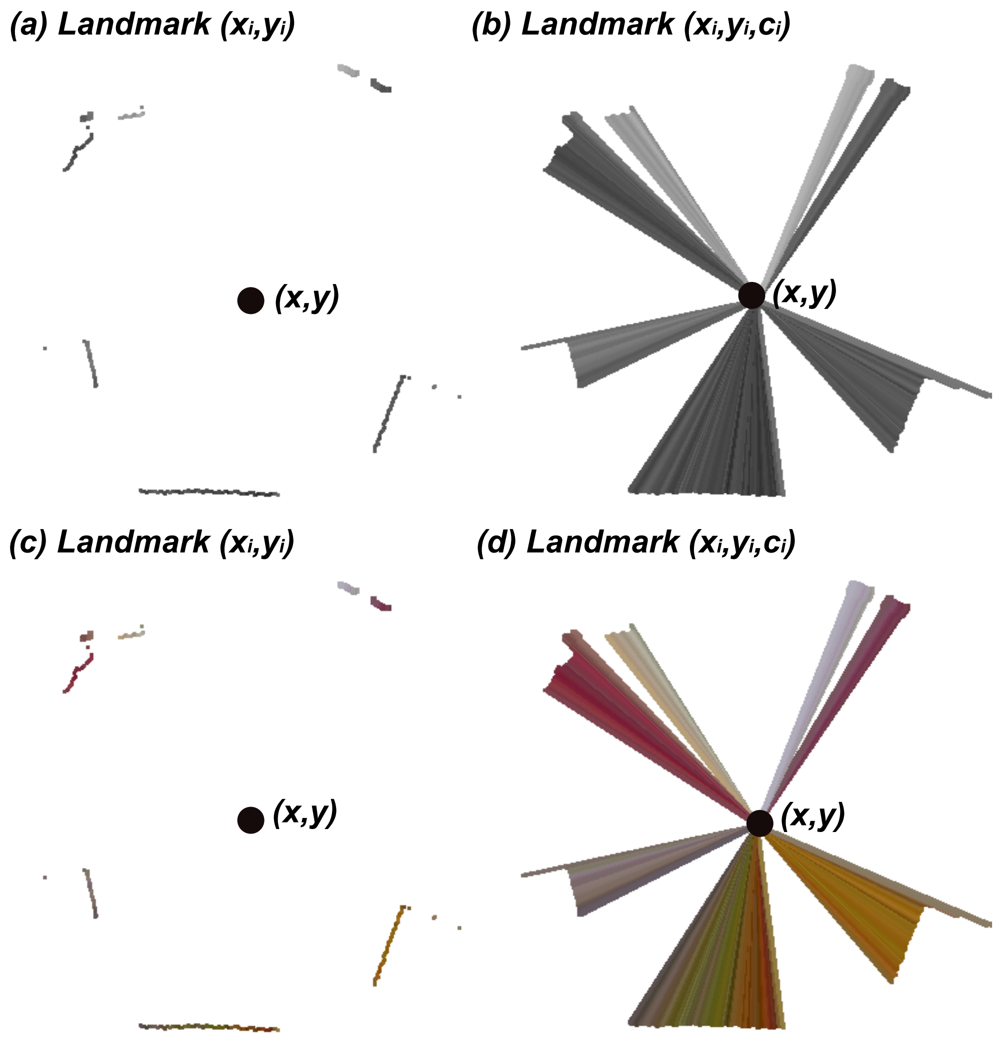


Figure 3.5: Difference between concepts. (a) is using only range with area integration method. (b) is using both range and color with area integration method. (c) is using only range with area integration method. (d) is using both range and color with area integration method.

where M for the number of objects, n for dimension number, C_j for feature value of j-th object, r_j for distance to feature and ρ'' for another normalize constant. In this part, we can check that the final output can be the same form of multidimensional case.

Figure 3.5 shows the differences between suggested methods. Our moment models can be divided into two parts. The one of generalized version (a) uses distribution of landmarks and we call it perimeter method. The other of generalized area integration (b) uses imaginary filled area in landmark distribution and call it area integration method. Also each one can be attached with color information like (c) and (d).

3.2 Experiments and Results

3.2.1 Simulations

3.2.1.1 Simulation about multi-dimension experiments

Initially we test our approach in simulation environment including a set of landmarks. A set of landmarks (discrete landmarks) are given in an arena and each landmark has its height and we assume that the distance to each landmark is obtained with a laser sensor. Figure 3.6, we can observe the unique convergence point with a feature value of height for each environment. The home location may be different from the convergence point but the moment model guides well the homing direction.

Figure 3.6 shows that the moment potential has a convex shape and it has the minimal peak point. Even if the landmarks are not uniformly distributed, the moment measure has the minimal reference point. A pair of positions including the home location can share the same reference point using the relative distances to a set of landmarks and the landmark features. In simulation environment, there is no error to estimate the reference point and the homing direction at each position is very accurate.

Also, we do some additional experiment about the location of convergence point according to the changes of order number (n). As calculation about the minimum point of moment function can be possible only when $n = 2$, we find the minimum points in other cases by area search. Figure 3.8 shows the movement of minimum point.The output is not linear.

3.2.1.2 Simulation about pixel-wise assumption

In this part, we test various cases with high dimension number having unique convergence that we proved. Different to prior case, we made simulation environment that can generate image-like measurement. These measurements have omni-directional range and color distribution of the horizontal line and it is very similar with real experiment. If our assumption is right, then this output can make successful outputs with higher dimension number.

Figure 3.9 shows some cases of using various dimension number. First one using $n = 2$ is equal to the normal case, third one using $n = 3$, fourth one using $n = 4$ and fifth

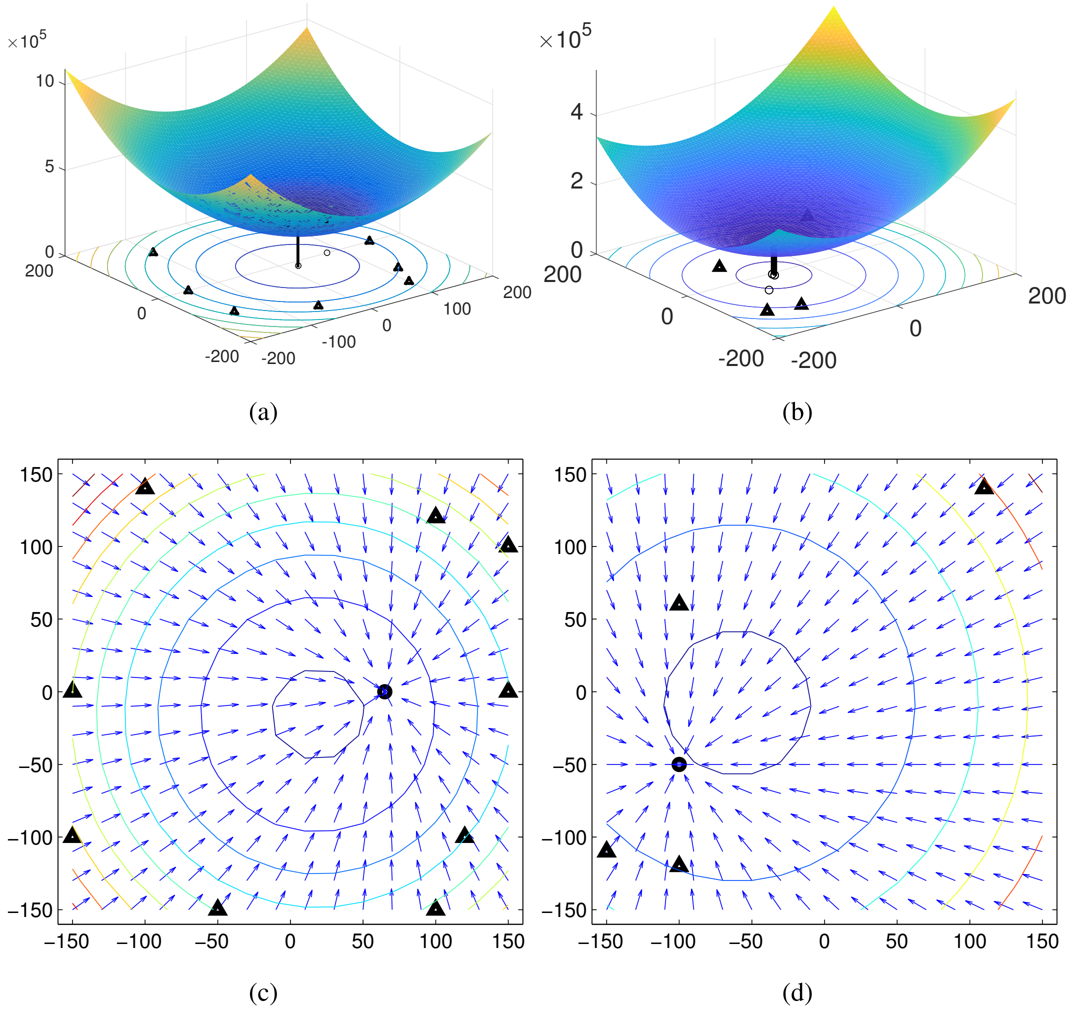


Figure 3.6: Test with two different simulation environments; the first row shows the moment potential and the second row shows vector maps including homing vector on the contour plot (triangles: landmark positions, black dot: home location)

one using $n = 10$. All the cases can make successful homing pattern and there are some differences. Therefore we can say that our generalized model can find homing direction with various dimension number.

Next, we check the homing performance according to changing dimension number to check our pixel-wise assumption. According our theorem, if the objects have cylinder shapes and ratio between distance to radius is enough small, homing performance using potential field with $n = 3$ has better output than prior one with $n = 2$.

Figure 3.10 shows the output and we can check that there is the best case when the dimension number is equal to 3. Also, there are red circles meaning statistically equal

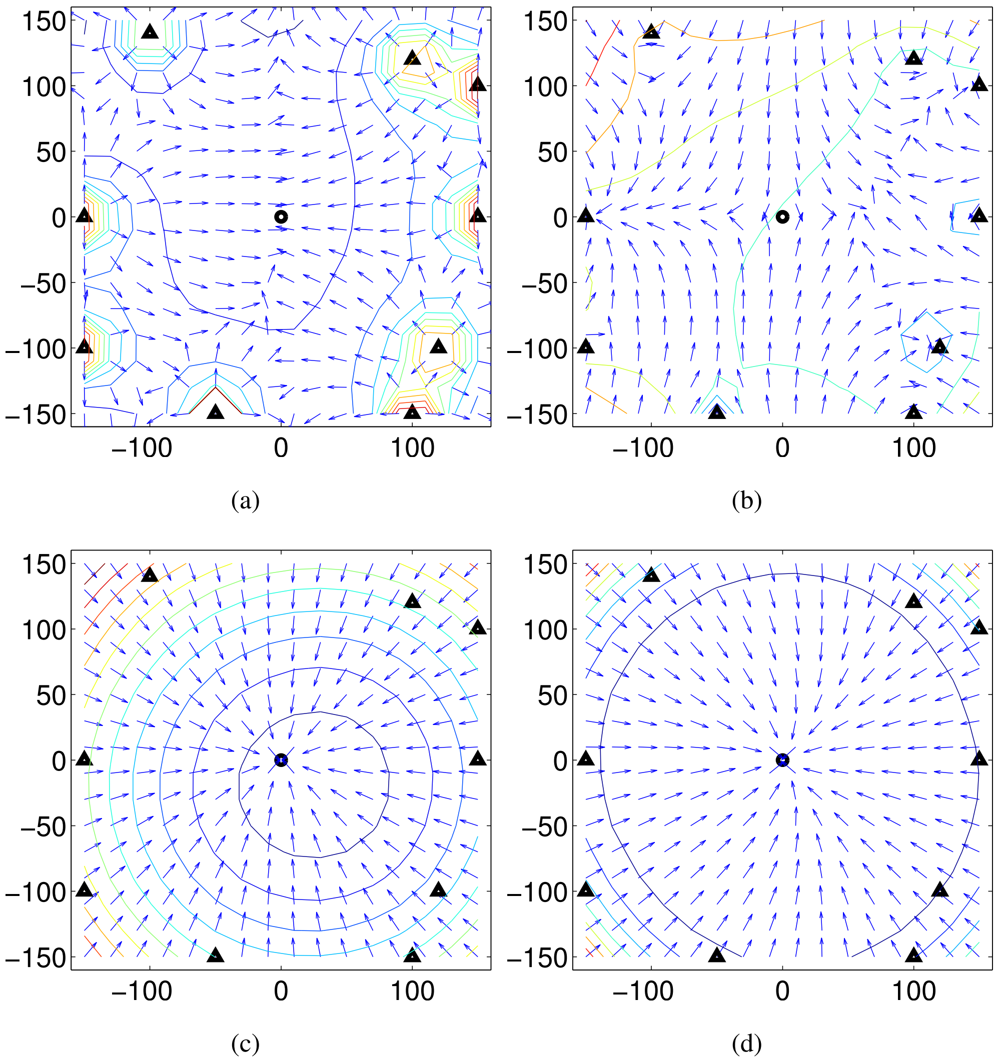


Figure 3.7: Simulation output for various moment functions. Vector maps including blue arrows pointing calculated homing direction at each position using $n = -1, 0.2, 2, 10$. Curved line shows the contour line of moment curve. Black dot shows the home location and triangle shows the location of landmark.

minimum points. Then we can check that minimum points is not equal to other cases and we can know that our assumption can work. However, this simulation is no measurement error and there is no background which also can be landmark and make occlusion.

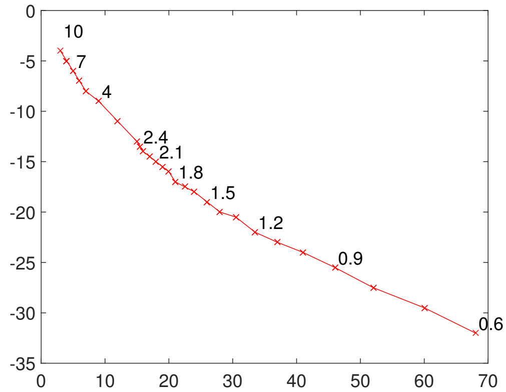


Figure 3.8: Experiment about the location of convergence point according to the changes of order number (n). The environment of experiment is equal to the ones of figure 11 and 12. Red dots are the minimum points with text figuring order number.

3.2.1.3 Simulation about inter-filled moment navigation

In the early part of this paper, we suggest the inter-filled moment navigation inspired from dark matter. It includes higher dimension term and, in prior parts, we check the convexity of moment function with various dimension numbers. Also we suggest generalized version having unique convergence with various dimension numbers. Through these parts with simulation environment, we can test our inter-filled moment navigation having three different dimension number.

Figure 3.11 shows some cases of using various dimension number in inter-filled model. Left one using $n = 2$ is equal to the normal case, right one using $n = 3$. In this simulation, those two cases make successful homing ability with very small differences. Therefore our generalized model can be applied into simulation with various dimension number of inter-filled moment model.

Next, we also check the homing performance according to changing dimension number to check our pixel-wise assumption in this part. Figure 3.12 shows that the output of 2 is the statistically best. Because combination of three dimension number makes the one ($n = 2$) using 2,3,4. Then we can think that global best ($n = 3$) based on pixel-wise assumption is partially applied into this part.

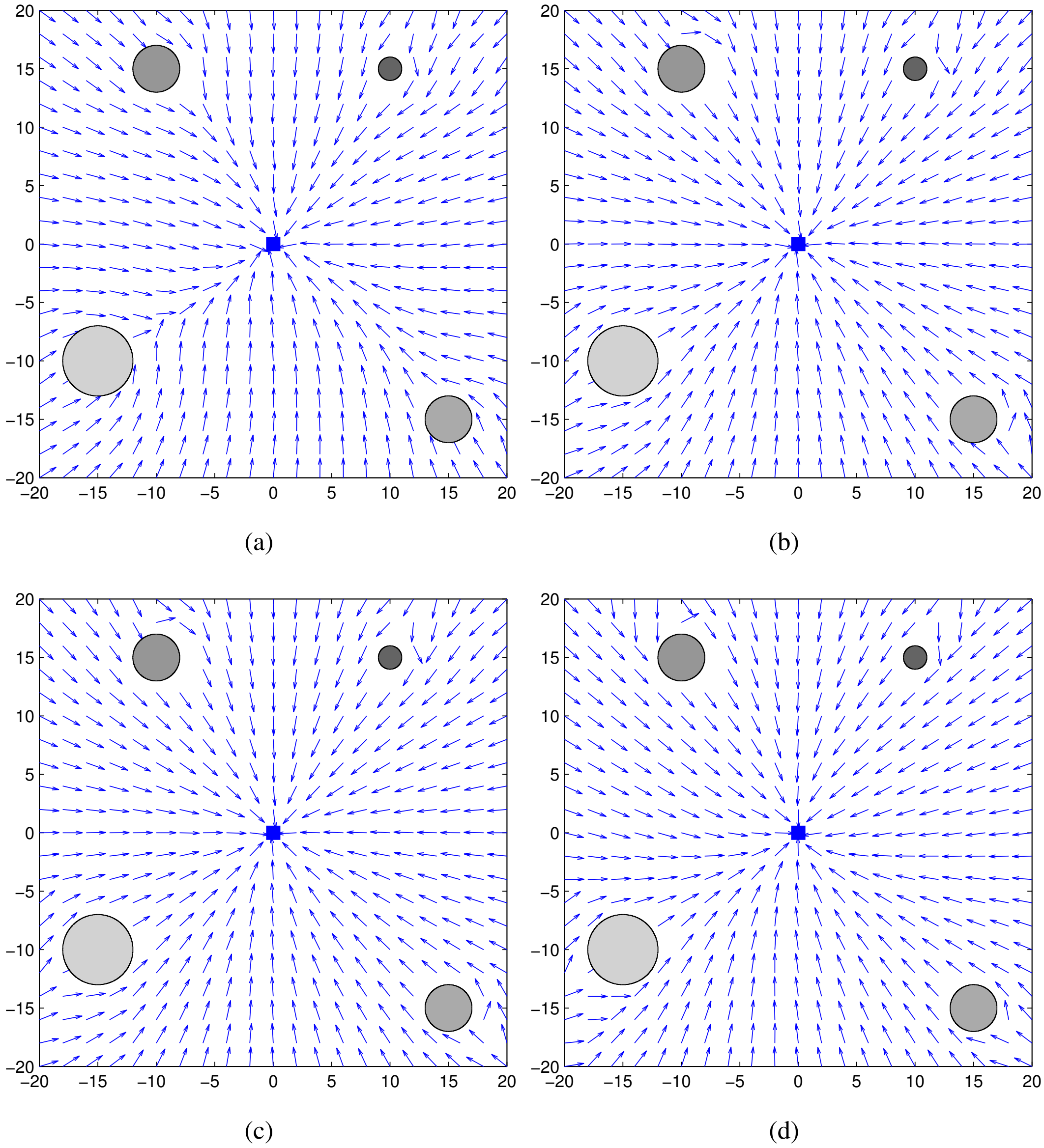


Figure 3.9: Simulation output for various moment functions having different dimension number in pixel-wise assumption. Vector maps including blue arrows pointing calculated homing direction at each position using $n = 2, 3, 4, 10$. Blue square shows the home location and four circles shows volume and color.

3.2.2 Experiments with Moment Model

Additional to output of our methods, we compare our second model with state-of-art techniques. In this part, we introduce some other algorithms which are used in comparison.

Table 3.1 shows various methods classified by the feature selection, the range sensor

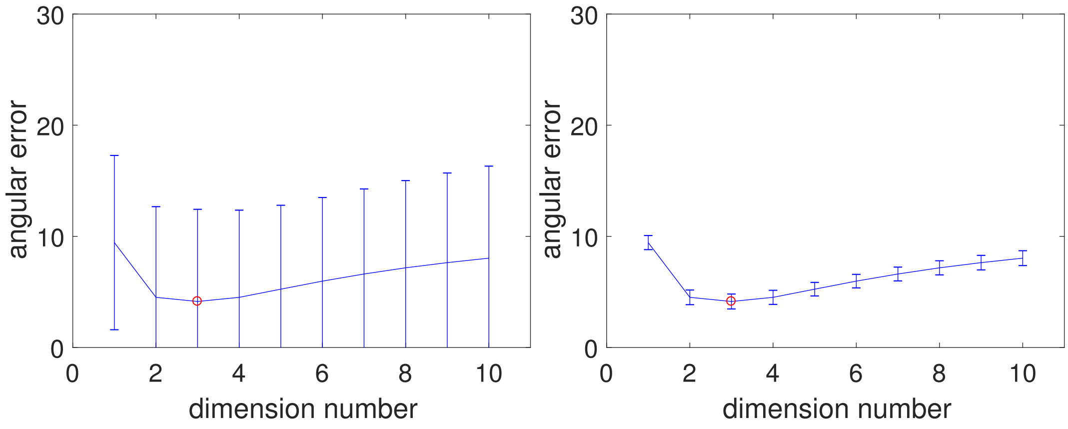


Figure 3.10: Performance graph for homing accuracy according to changing dimension number (n) of generalized moment function. The environment of experiment is equal to the ones of Figure 3.9. Red dots means the minimum point and statistically equal point.

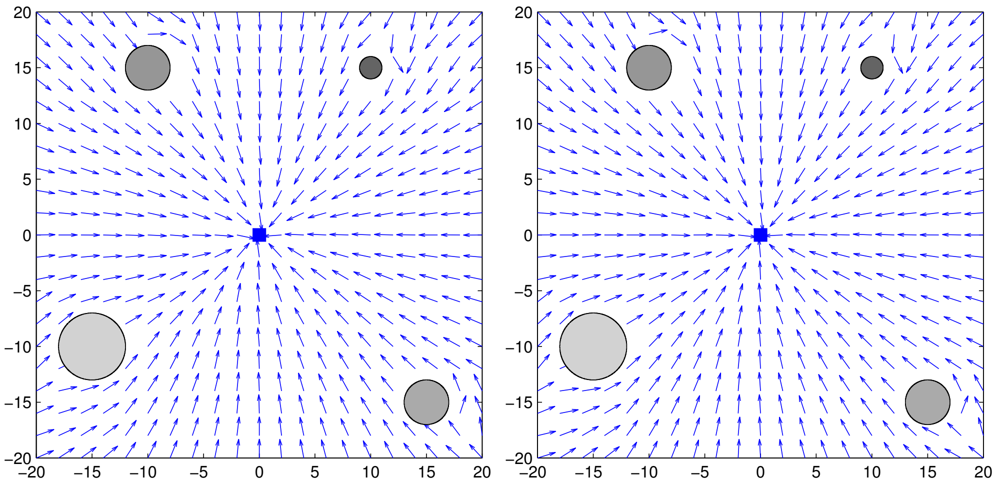


Figure 3.11: Simulation outputs for inter–filled moment functions. Vector maps including blue arrows pointing calculated homing direction at each position using $n = 2, 3$. Blue square shows the home location and four circles shows volume and color.

or distance estimation, and the alignment process. With reference compass, no alignment process is required, and six methods are available. Without reference compass, eight methods are listed (for $C_i = 1$, no visual feature is observed and only landmark rearrangement will be tested, since the visual compass is not applicable). In the result part, we try to check overall performances of these combinations.

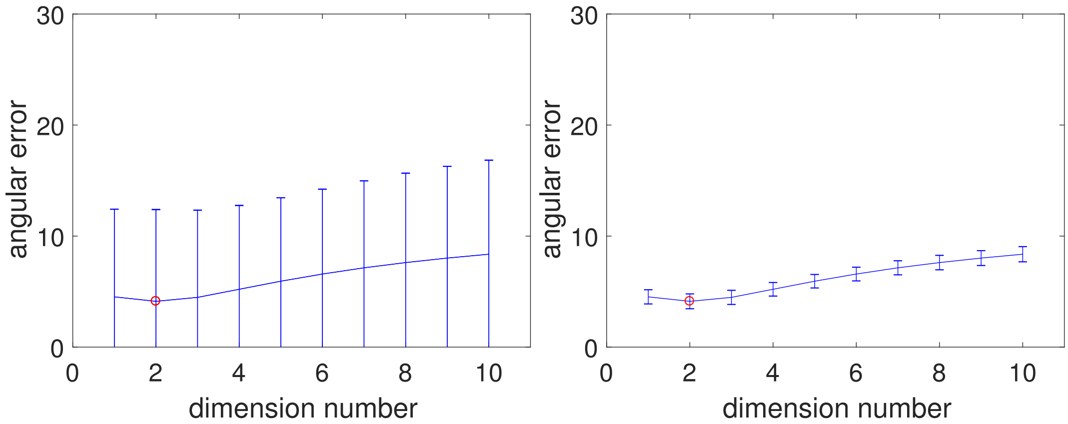


Figure 3.12: Performance graph for homing accuracy according to changing dimension number (n) of inter-filled moment function. Red dots means the minimum point and statistically equal point.

Table 3.1: Various methods classified by the feature, the range sensor and the alignment process

	Method	Feature	Range	Alignment method	Class
with reference compass	DID	color intensity	no sensor	aligned by compass	I
	moment	color intensity	equal distance ($r_i = 1$)	aligned by compass	II
	moment	equal intensity	ground-line estimation	aligned by compass	III
	moment	color intensity	ground-line estimation	aligned by compass	IV
	moment	equal intensity	laser sensor	aligned by compass	V
	moment	color intensity	laser sensor	aligned by compass	VI
without reference compass	DID	color intensity	no sensor	visual compass	I_v
	moment	color intensity	equal distance ($r_i = 1$)	visual compass	II_v
	moment	equal intensity	ground-line estimation	landmark rearrangement	III_r
	moment	color intensity	ground-line estimation	visual compass	IV_v
	moment	color intensity	ground-line estimation	landmark rearrangement	IV_r
	moment	equal intensity	laser sensor	landmark rearrangement	V_r
	moment	color intensity	laser sensor	visual compass	VI_v
	moment	color intensity	laser sensor	landmark rearrangement	VI_r

3.2.2.1 Moment model in real environment with depth sensor readings

We investigate a possibility of the moment model in real environments. The moment measure allows various features for landmarks in its calculation. In our experiments, a holistic approach over the landmark distribution is used and the omnidirectional depth information in combination with visual features are encoded in the moment value.

Initially we measured the moment with RGB color intensities in the panoramic image. The holistic view take all the pixels as landmarks in the environment. The information about 720 pixels near the horizontal line and the corresponding depth determines a mo-

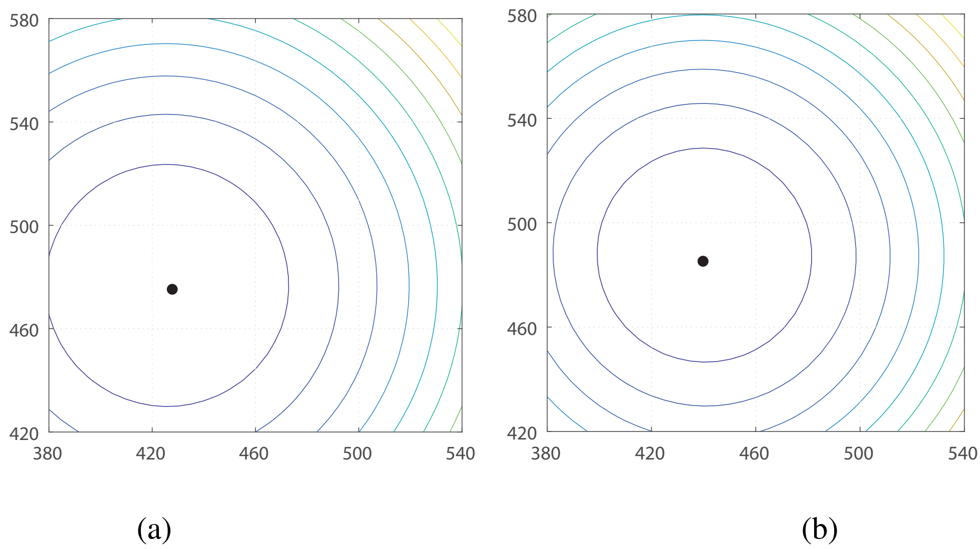
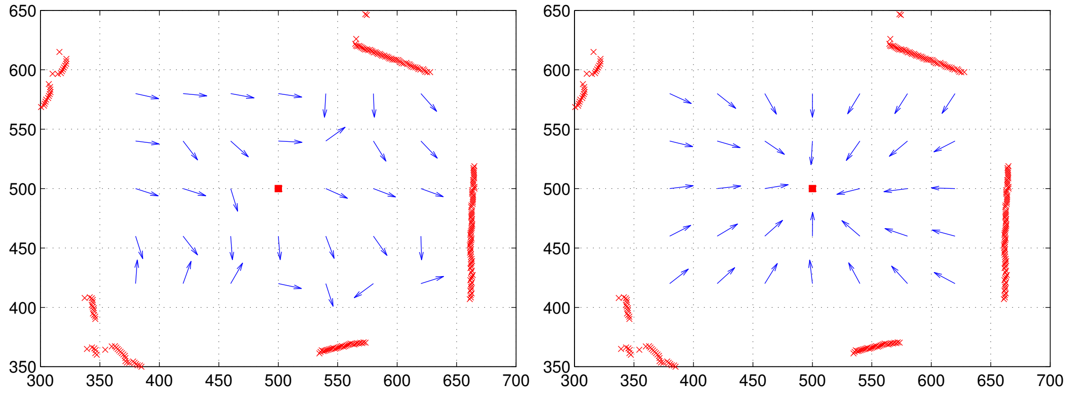


Figure 3.13: Contour plots at two different positions in the indoor environment (a) contour of moment potential at (460, 460) (b) contour of moment potential at home position (500, 500); black dots indicate the minimum potential points

ment potential. The contour map of the moment potential is displayed in Figure 3.13. The range sensor readings and color intensities can change depending on the observation point. Figure 3.13 shows that the minimal potential positions closely match each other in real environment, although there is a little deviation of the minimal potential position (reference point) when it is calculated at two different observation points. The measurement error or a few occlusions of landmarks can induce a little deviation of the ideal reference point, thus causing homing error potentially.

To see the effect of the two components, the color intensity and the distance of landmarks, we tested the moment with the unit distance $r_i = 1$ or the unit feature $C_i = 1$. In the first test, only color intensity is available for landmarks under equal distance assumption (unit distance $r_i = 1$). Figure 3.14 (a) shows the result and relatively large homing errors are observed. It implies that landmarks only with color intensity have difficulty in representing the surrounding environment. The distribution of RGB colors change depending on the observation point, which will highly influence the estimation of the reference point as well as the homing direction. In contrast, another test with continuous-ranged distance but no color intensity shows much better homing performance – see Figure 3.14 (b). The depth information of landmarks greatly contributes to the estimation of homing direction.



(a) (b)

Figure 3.14: Homing performance only with color intensity or only with range data (reference compass available) (a) only with vision but the unit distance ($r_i = 1$) (all landmarks have equal distances) (b) with only range data but the unit feature ($C_i = 1$) (all landmarks have the same visual feature or same color intensity) (red box at (500, 500) indicates home and arrows show the homing direction at each point)

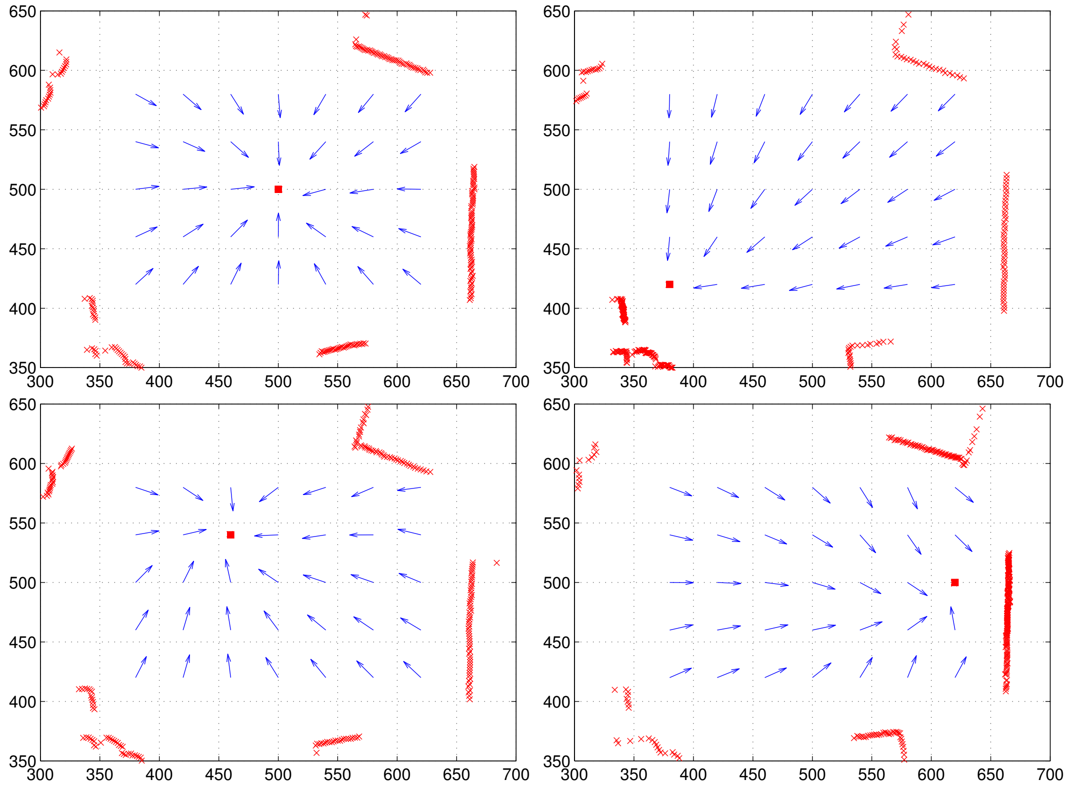


Figure 3.15: Homing performance with both range data and color intensity (reference compass available); red box indicates home position and the arrows show the homing direction at each point

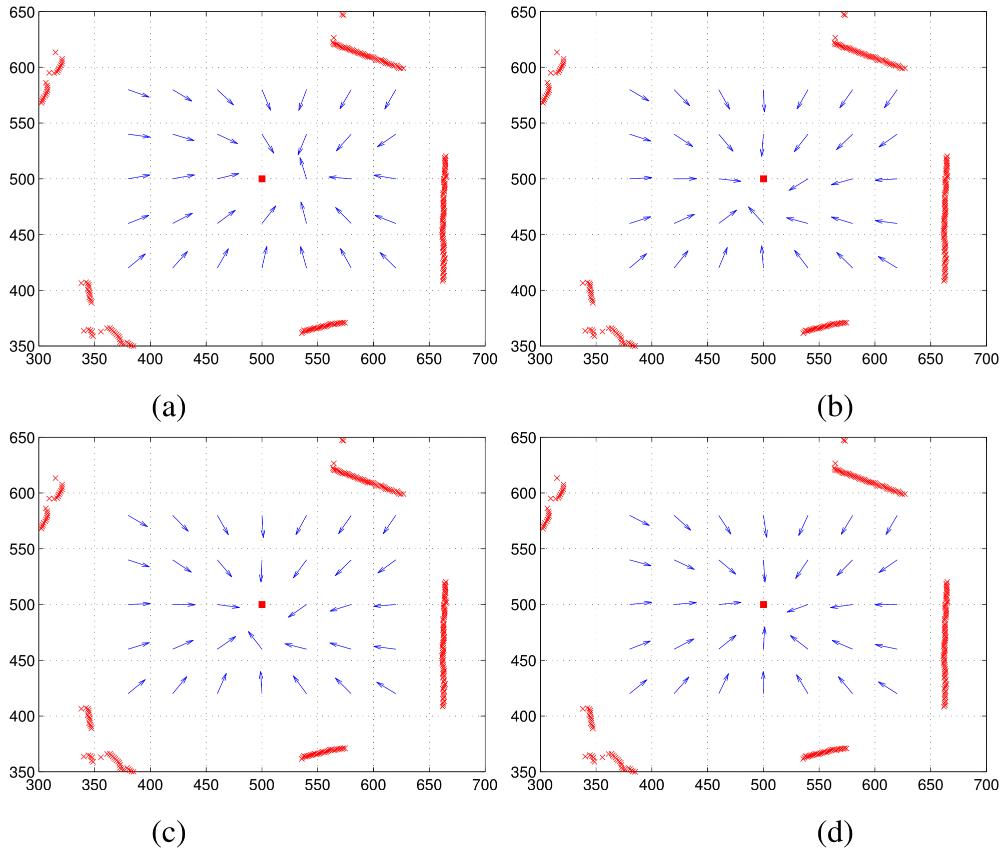


Figure 3.16: Homing performance with various feature values (a) C_i is the blue color intensity (b) C_i is the red color intensity (c) C_i is the V intensity in HSV space (d) $C_i = 1$ or 0 depending on the discetized condition with HSV (if $\sqrt{H^2+S^2+V^2} > 50$, $C_i = 1$, otherwise $C_i = 0$); red box at (500, 500) indicates home and arrows show the homing direction at each point.

Figure 3.15 shows vector maps using both the distance information and color intensity when a reference compass is available. The moment model with both features shows good homing performance at any position in the environment, mostly better than the moment model only with distance. Thus, the moment model, a combination model of distance and visual features, is a more promising approach to read the landmark distribution or characteristics in the environment.

Instead of the RGB space, we can apply one color, for example, red-colored intensity to the moment measure. In this case, a single reference point is available to derive the homing vector. Also, the HSV space for one pixel can produce another visual feature and the moment with the HSV intensity can derive different reference points, but it also produces successful homing performance. To discretize the continuous-ranged

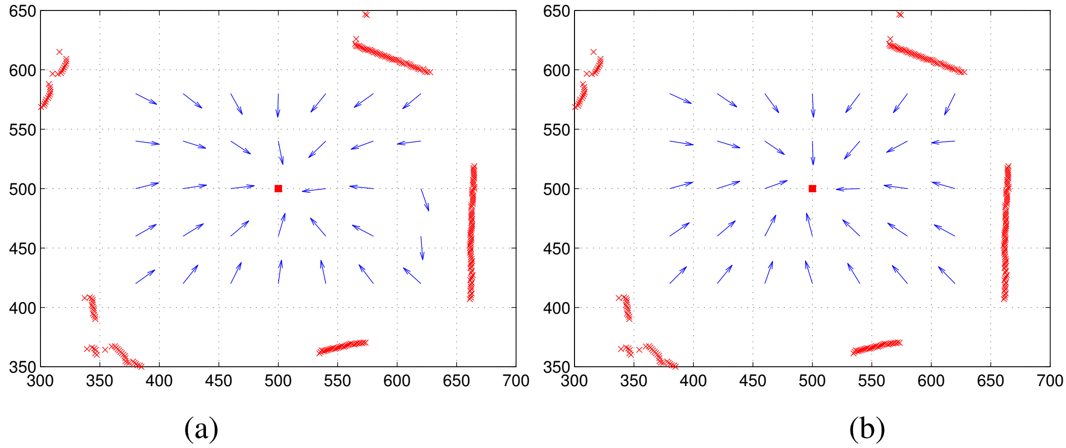


Figure 3.17: Homing performance of moment model without reference compass; both range data and color intensity are used and the coordinate alignment process is applied (a) visual compass (b) landmark re-arrangement method

attribute value, a cut-off threshold can be used; if a feature value is greater than the threshold, we set $C_i = 1$, otherwise $C_i = 0$. In this case, the moment measure is like extracting a special feature of landmarks, by collecting all the landmarks with $C_i = 1$. Then the moment model can be converted to the ALV or the DELV model depending on the distance information (unit distance or continuous-ranged distance). Figure 3.16 shows the homing performance result depending on various patterns of the feature C_i. Even one color intensity together with depth information provides reasonable homing performance. More features or a different representation of the visual feature like the HSV space can estimate well homing directions. A choice of special features satisfying a given condition can also find homing directions – see Figure 3.16(d).

If there is no reference compass available, we need to align the coordinates for the snapshot images obtained at two positions. One of popular approaches is the visual compass approach by pixel matching process. Another approach is the landmark rearrangement method. The methods were applied to the above environment. The moment model priorly needs this alignment process. Figure 3.17 shows the results. With visual compass, there are some large angular errors at the right side corner, while the landmark re-arrangement method shows relatively small angular errors towards home at all the points.

To compare the moment model and the DID (Descent in Image Distance) method, we demonstrated the homing results with the DID method in Figure 3.18. The DID method uses only visual images to estimate the homing direction. Despite the limitation, it can

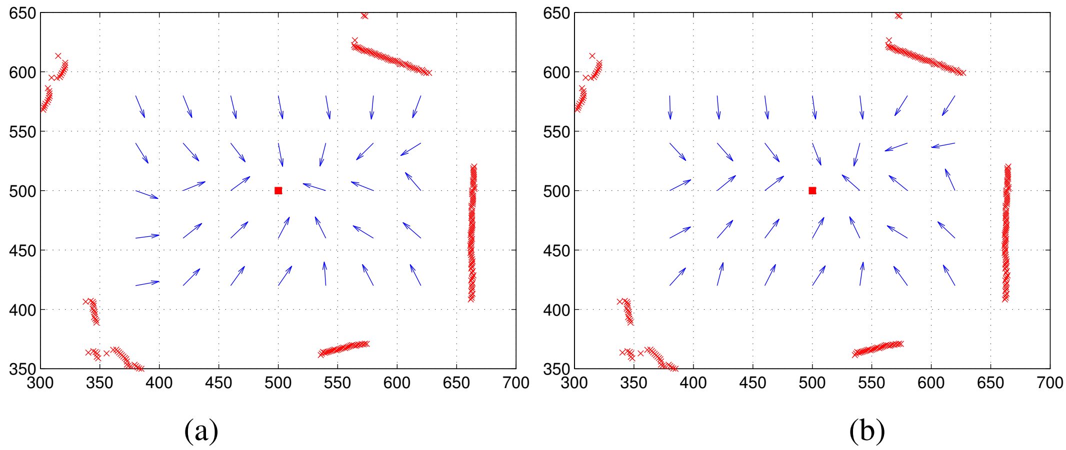


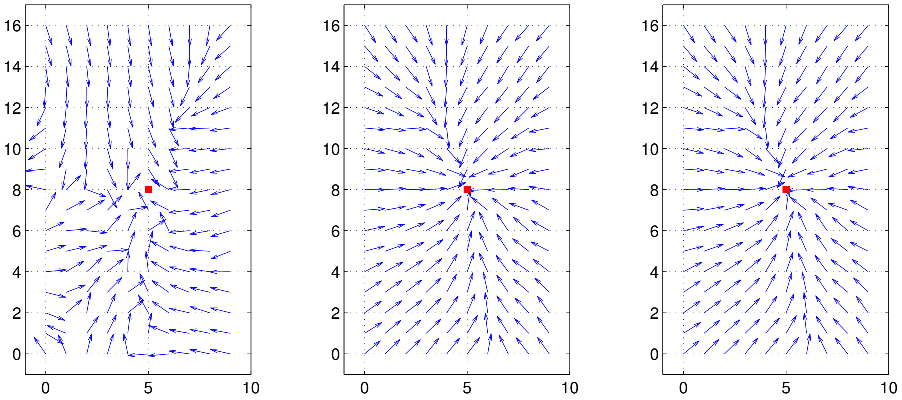
Figure 3.18: Homing performance with the DID approach (a) with reference compass (b) visual compass without reference compass; red box indicates home position and arrows show the homing direction at each point.

estimate the homing directions reasonably. It seems that whether the DID method has reference compass or not has no large impact on the homing performance. For overall, the DID method has larger angular errors in the average than the moment model, and it is due to lack of depth information of landmarks.

3.2.2.2 Moment model in real environment with ground-line distance

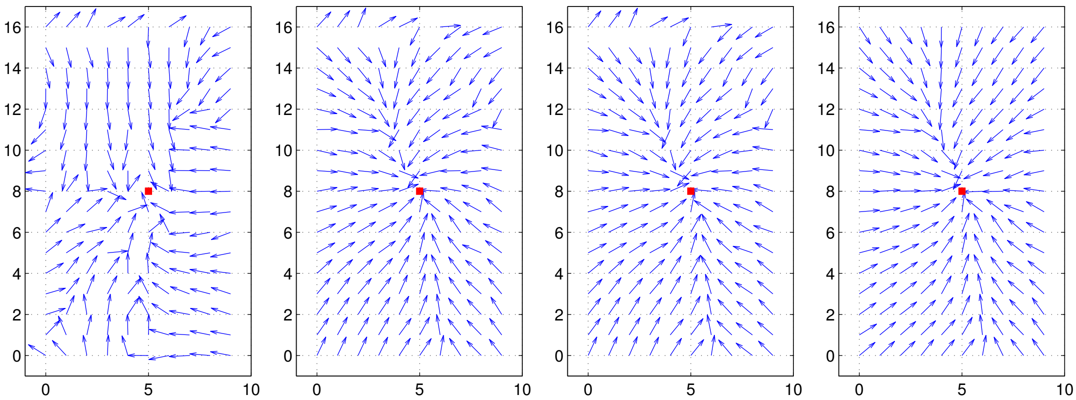
We applied our moment model to another environment, one of Vardy's dataset called 'a1original' (Möller et al., 2007), which include many panoramic snapshots, but without distance map. The omnidirectional images were collected for the indoor environment and the arena size is 2.7m x 4.3m and there are 170 points with 10 cm regular intervals for snapshots.

Figure 3.19 shows the homing performance results with reference compass. Similar to the previous results with our lab environment *env*0 shown in figure 3, the moment model only with color intensity shows the worst performance. The model with ground-line distance shows very successful performance. The model with both distance and color intensity together shows a little more improved performance. Some positions have angular deviation in the direction to the goal position, which is related to the error in distance estimation. Without laser sensor, only visual image can determine the homing direction effectively. It can be inferred that the distance parameter in the moment model greatly influences the homing performance.



(a) (b) (c)

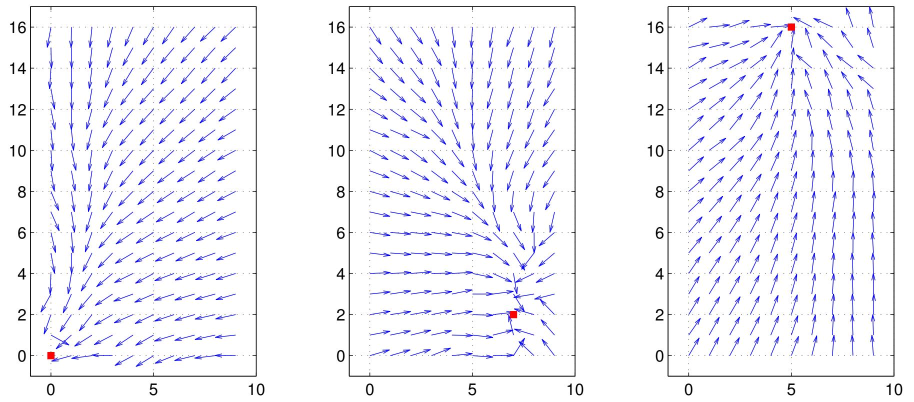
Figure 3.19: Homing performance for Vardy's image environment with reference compass (a) moment model only with color intensity (b) moment model only with estimated distance (c) moment model with estimated distance and color intensity; distance is estimated using the ground line in the image without laser sensor (arrow indicates the homing direction)



(a) (b) (c) (d)

Figure 3.20: Homing performance for Vardy's image environment without reference compass (a) visual compass and moment model only with color intensity (b) visual compass and moment model only with estimated distance (c) visual compass and moment model with both estimated distance and color intensity (d) landmark rearrangement method and moment model with both estimated distance and color intensity; distance is estimated using the ground line in the image without laser sensor (arrow indicates the homing direction)

Figure 3.20 shows the performance depending on the coordinate alignment methods. Even without reference compass, similar patterns of homing performance were observed. The moment model only with color intensity is insufficient to guide homing,



(a) (b) (c)

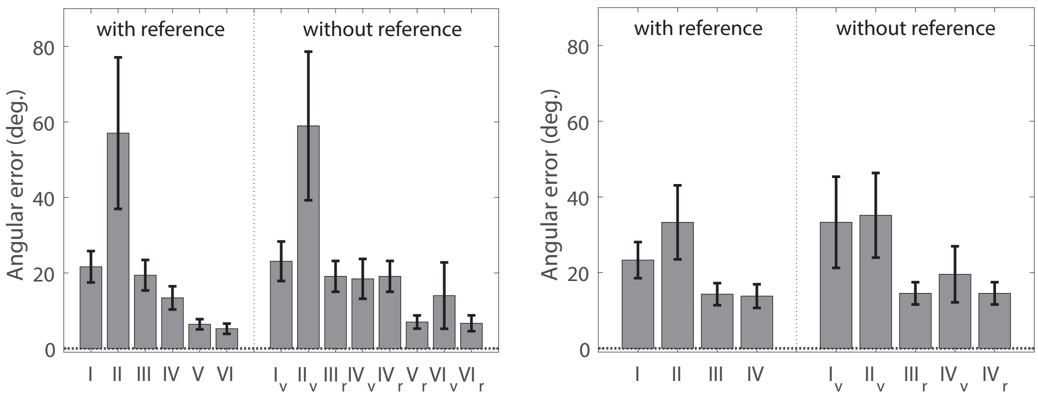
Figure 3.21: Homing performance for Vardy's image set without reference compass; the landmark rearrangement method to align the coordinate and the moment model with estimated distance and color intensity are used (a) home location (0,0) (b) home location (7,2) (c) home location (5,16)

but the model with ground-line distance shows much better homing results. Also, the landmark re-arrangement method seems better than the visual compass method – see Figure 3.20 (c)-(d). Even when we change the home locations, the moment model with landmark rearrangement robustly works well for homing as shown in Figure 3.21. With only visual snapshots, we are successful to estimate the distance to landmarks as well as adjust the coordinate alignment, and to decide the homing direction ultimately.

3.2.2.3 Comparison of homing performances with various methods

We measured angular errors for homing directions with various methods listed in Table 3.1. We assume that the desired homing direction is the direct path from the current position to the home location. In the moment model, we encoded the depth information of landmarks and the visual features into a moment potential. We also tested the coordinate alignment for the environment without reference compass. The snapshot image at the home location becomes warped at the current position and it influences the landmark distribution and the moment characteristics.

The method *I* and *II* use only visual features to decide the homing direction and they have large angular errors. The moment model without depth information has shortcomings, compared to the DID method (method *I*) as we see the performance for the method *I* and *II*. However, in the moment model, the depth information of landmarks



(a) (b)

Figure 3.22: Homing performance with reference compass and without reference compass; x-axis indicates testing methods described in Table 3.1 (a) angular errors in the indoor environment (b) angular errors in Vardy's image set

by a laser sensor or the estimation of the ground-line distance in the visual image significantly improves the homing performance – see the method $III - VI$. The same result can be found in both the indoor environment *env0* and Vardy's data set. Also, if there is a real measurement of the distances to landmarks, it can improve further the performance as shown in Figure 3.22 (a).

We also found that the landmark rearrangement is quite effective in aligning the coordinates. In many cases, the average performance with the landmark rearrangement is better than that of the visual compass. The methods (IV and VI) with a combination of two components, landmark distance and visual feature, has lower error performance, when compared to the methods with only distance (III and V). The DID method is based on the visual images and their image difference, and the above experiments imply that only visual information seems insufficient to guide homing. The DID is worse in homing performance than the moment model with the visual feature and the depth information.

Table 3.2-4.2 shows angular errors for homing direction. Mostly the errors are within 45 degrees. The method II only with the visual feature shows a relatively large portion of angular errors greater than 45 degrees in the two different environments. The results are consistent with the performance shown in Figure 3.22. Especially, the DID method shows somewhat poor performance in Vardy's image set without reference compass.

Table 3.2: Angular errors with various methods for the indoor environment *env0* (N is the number of testing points)

method		error (deg.)	N	$0 \leq \varepsilon_\theta < 45°$	$45 \leq \varepsilon_\theta < 90°$	$90 \leq \varepsilon_\theta < 180°$
with reference compass	I	21 (±12)	34	97.1%	2.9%	0%
	II	57 (±57)	34	67.6%	5.9%	26.5%
	III	19.4 (±11.6)	34	97.1%	2.9%	0%
	IV	13.4 (±8.8)	34	100%	0%	0%
	V	6.4 (± 4.3)	34	100%	0%	0%
	VI	5.7 (±4.1)	34	100%	0%	0%
without reference compass	I_v	23 (±15)	34	94.1%	5.9%	0%
	II_v	59 (±56.4)	34	58.8%	14.7%	26.5%
	III_r	19.1 (±11.7)	34	97.1%	2.9%	0%
	IV_v	18.4 (±15.1)	34	94.1%	5.9%	0%
	IV_r	19.1 (±11.7)	34	97.1%	2.9%	0%
	V_r	7 (±5)	34	100%	0%	0%
	VI_v	14 (±25)	34	94.1%	0%	5.9%
	VI_r	7 (±6)	34	100%	0%	0%

Table 3.3: Angular errors with various methods for Vardy's image set (N is the number of testing points)

method		error (deg.)	N	$0 \leq \varepsilon_\theta < 45°$	$45 \leq \varepsilon_\theta < 90°$	$90 \leq \varepsilon_\theta < 180°$
with reference compass	I	23.31 (±13.66)	169	97.1%	2.9%	0%
	II	33.28 (±28.01)	169	72.19%	24.26%	3.55%
	III	14.31 (±8.35)	169	100%	0%	0%
	IV	13.8 (±8.96)	169	100%	0%	0%
without reference compass	I_v	33.30 (±34.54)	169	79.88%	11.83%	8.29%
	II_v	35.15 (±32.02)	169	69.82%	23.08%	7.1%
	III_r	14.53 (±8.43)	169	100%	0%	0%
	IV_v	19.56 (±21.22)	169	96.45%	0%	3.55%
	IV_r	14.53 (±8.43)	169	100%	0%	0%

Table 3.4: Table for various types of experiments

	feature	range	alignment	Dimension Number	mark
With Compass	Color Intensity	None	Aligned	none	I
	Equal Intensity Assumption	Estimation	Aligned	2	II
	Intensity	Estimation	Aligned	2	III
	Equal Intensity Assumption	Estimation	Aligned	4	IV
	Color Intensity	Estimation	Aligned	4	V
	Equal Intensity Assumption	Measurement	Aligned	2	VI
	Color Intensity	Measurement	Aligned	2	VII
	Equal Intensity Assumption	Measurement	Aligned	4	$VIII$
	Color Intensity	Measurement	Aligned	4	IX
	Equal Intensity Assumption	Measurement	Aligned	4 -numerical	X
Without Compass	Color Intensity	None	Visual Compass	none	I_v
	Color Intensity	Measurement	Visual Compass	2	VII_v
	Color Intensity	Measurement	Rearrangement	2	VII_r
	Color Intensity	Measurement	Visual Compass	4	IX_v
	Color Intensity	Measurement	Rearrangement	4	IX_r

3.2.3 Experiment about Various Dimension Number

Based on our model, we made some combinations to build various navigation models. Following Table 3.4,there are 14 cases. With reference compass (aligned), there are three cases. The first method is using DID (I) as reference, second ones of moment model using equal intensity assumption and estimated distances with perimeter method (II^P) and third with both color intensities and estimated distances with perimeter method (III). Fourth is with area integration method (II^A) without color and fifth area integration method (V) with color. The ones from sixth to ninth use measured distances instead of estimated distances ($VI, VII, VIII, IX$). To compared with our derivative model with numerical method, we add one more type using numerical searching (X) for convergence point when the dimension number is four.

Without reference compass, there are five variations by combination of various alignment algorithm with both DID and moment model. First one is using DID after alignment by visual compass (I_v). Second method are moment models using intensity and measured distances with perimeter method aligned with visual compass (VII_v), rearrangement method (VII_r). Third methods are with area integration method aligned with visual compass (IX_v), rearrangement method (IX_r). In the result part, we try to check overall performances of these combinations.

In this part, we do experiment about the effect about dimension number of moment function navigation. We use laser sensor to get range data and vision sensor to color information. Through the experiments above, we can check the homing convergence. In those experiments, they can almost perfectly find the homing direction at each current position and the outputs have small differences. Also, we do experiments for generalized moment form ($n = 0.1, 2, 4, 6, 8, 10$) in this real environments.

3.2.3.1 Robot experiment for multi-dimension tests

Next, we apply our model to real environment. We do robotic experiment based on simulation output with environment using MATLAB 2017a. This environment, as mentioned above, have various objects like dresser, drawers, trash cans, large vases, windows, partitions as landmark objects. Then our vision and range sensors make panoramic measurement and we combine them by using generalized moment model.

First, we do experiment about the effect about dimension number of moment function

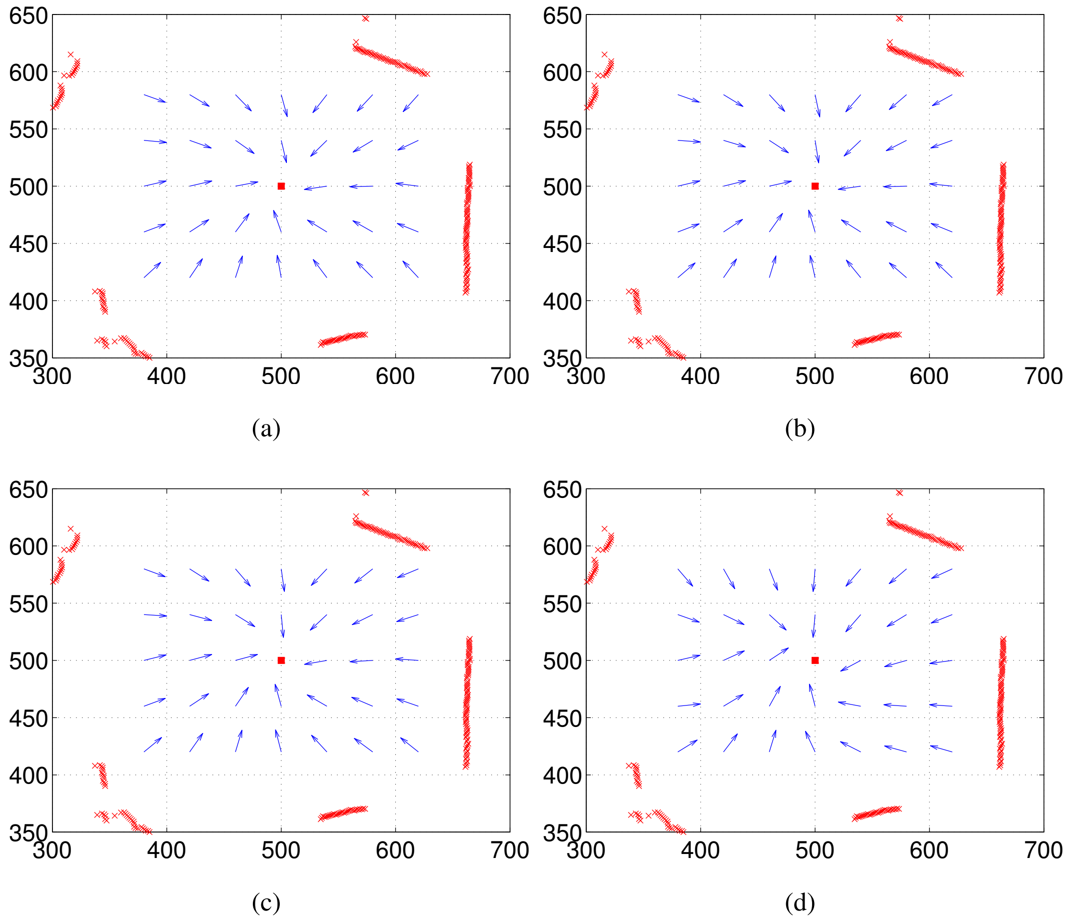


Figure 3.23: Robot experiment output for various moment functions. Vector maps including arrows pointing homing direction use $n = 2, 3, 4, 10$. Environment is equal to the prior figure. The red box at (500,500) is home and the arrows show the homing direction at each point.

navigation. We use laser sensor to get range data and vision sensor to color information. Through the experiments above, we can check the homing convergence. In those experiments, they can almost perfectly find the homing direction at each current position and the outputs have small differences. Also, we do experiments for generalized moment form ($n = 0.1, 2, 4, 6, 8, 10$) in this real environments. Figure 3.23 is the output and we can check the homing convergence in cases whose n variable over 2.

Then we can check that these cases using real data can also make successful homing pattern and there are some differences. All those outputs agree with the ones of simulation. Therefore we can say that our generalized model can find homing direction also in real experiment with various dimension number.

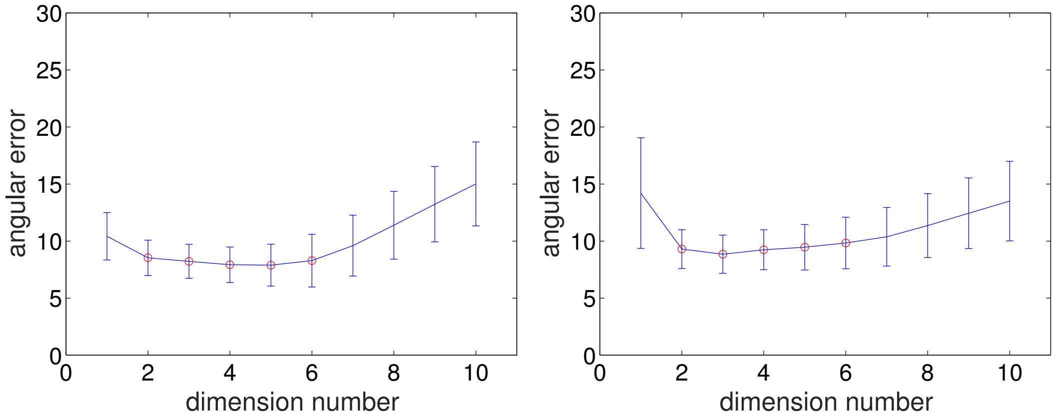


Figure 3.24: Performance graph for homing accuracy according to changing dimension number (n) of generalized moment function. Left one is without color information and right one is with both color and range. Red dots means the minimum point and statistically equal point.

Next, we check the homing performance according to changing dimension number. Figure 3.24 shows the output and we can check that there is best case when the dimension number is equal to 3 (not 2). However, there are red circles meaning statistically equal minimum points and we cannot easily say that the one using $n = 3$ is the best. We think that our assumption is not strictly satisfied in real world then the output has weak effect by higher dimension.

3.2.3.2 Robot experiment for various forms

In this part, we did some tests using various form of moment functions.

I, I_v : DID First of all, in this part, we analyse other algorithm called DID. It requires multiple images as references for finding homing direction and Figure 3.18 shows the output using three images. These outputs also successful homing patterns with compass or without compass (in this part, we use visual compass in alignment). Figure 3.25 shows the output using DID. First one is with compass sensor and second one with visual compass algorithm.

II^P and II^A: moment methods using estimated distances

In this part, we apply the estimate distances to moment model. Figure 3.26 shows the outputs. These models use estimated distances. First one (II^P) uses generalized moment model with perimeter without color information and second one (III) with

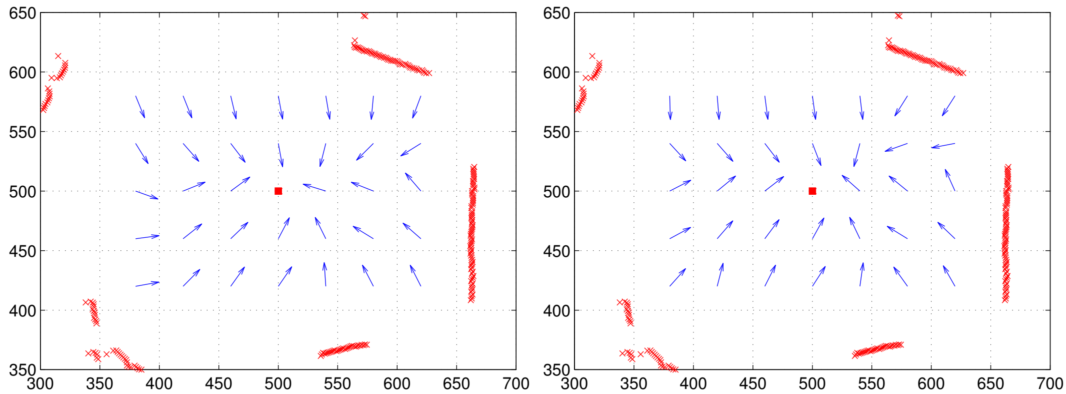


Figure 3.25: DID result using three visual cues as reference images. Left one is for aligned condition and right one is using visual compass. The red boxes at (500,500) are home position of those experiments and the arrows show the homing direction at each point. The left one is using compass sensor and the second with visual compass method in alignment.

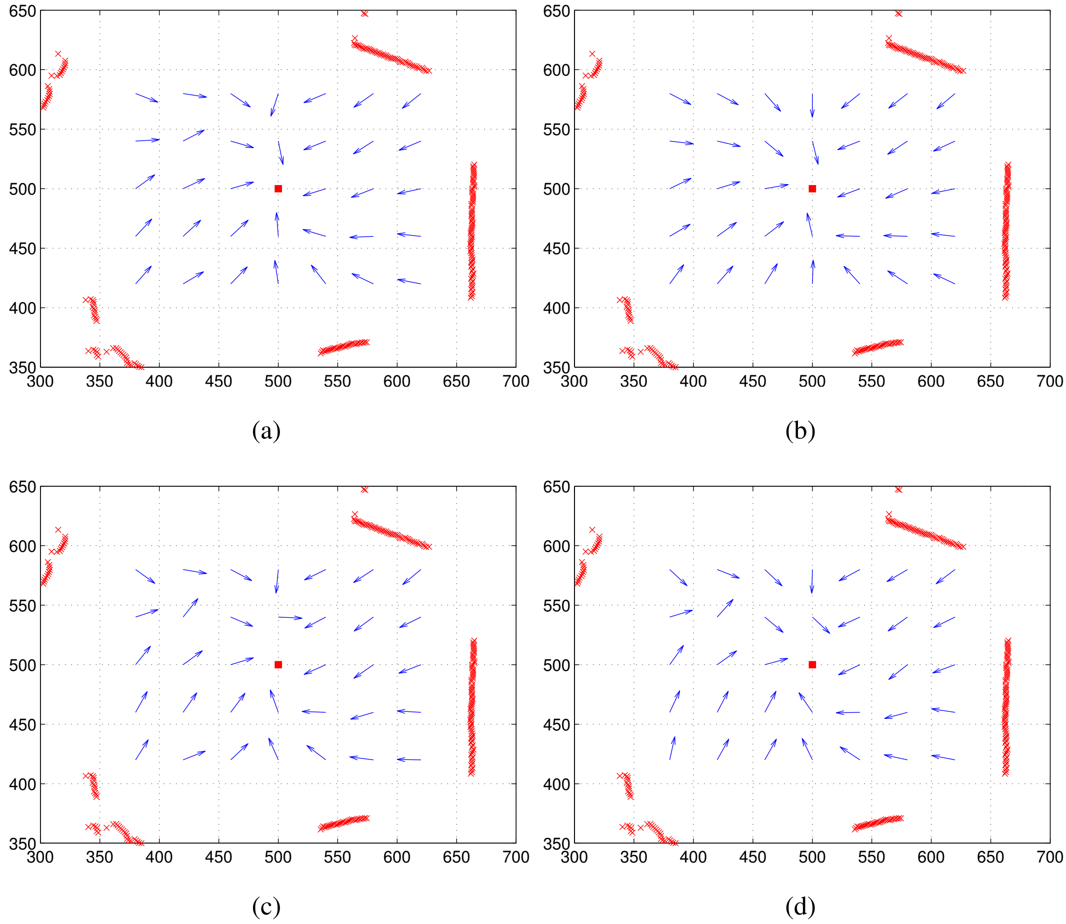
both range and color intensities. Third one (IV) uses generalized moment model with area integration method without color and fourth one (V) with color.

III^P and III^A: moment methods using measured distances

In this part, we apply the estimate distances to moment model. Figure 3.27 shows the outputs. These models use measured distances. First one (III^P) uses generalized moment model with perimeter without color information and second one (VII) with both range and color intensities. Third one ($VIII$) uses generalized moment model with area integration method without color and fourth one (IX) with color.

VII_v, IX_v, VII_r, IX_r: using both measured distances and intensities in non-aligned condition

Similar with experiments above, we test our models without compass. These models use measured distances and color intensities. Figure 3.28 shows the outputs. First one (VII_v) uses generalized moment model with perimeter and second one (IX_v) uses area integration model with visual compass. Third one (VII_r) uses generalized moment model with perimeter method and fourth one (IX_r) uses area integration model with rearrangement algorithm.

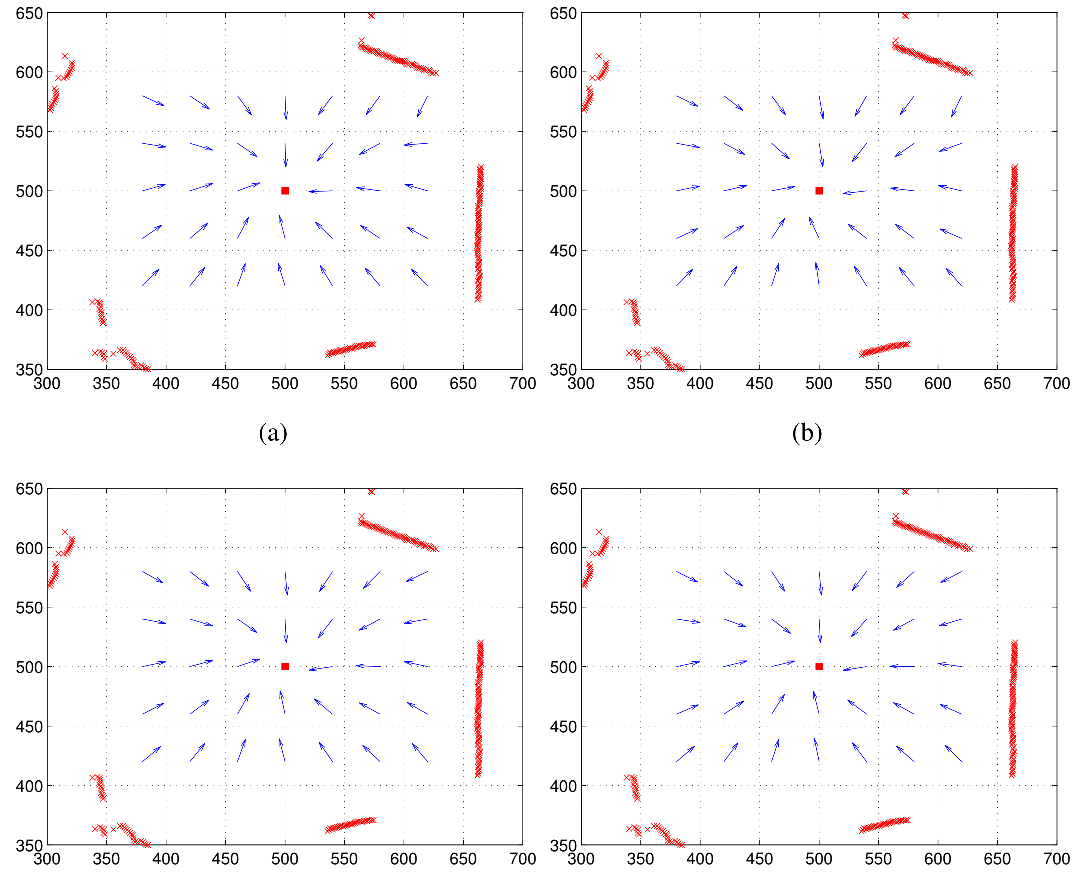


(a) (b) (c) (d)

Figure 3.26: Robot experiment result with estimated distances. Using perimeter moment model : (a) shows the case without color, (b) with estimated distances. Using area integration moment model : (c) without color, (d) with estimated distances. The red box is home located at (500,500) and the arrows show the homing direction at each point.

3.2.3.3 Comparison with numerical method(X)

In this part, we test our two methods. The one is based on derivative of moment curve with area integration and the other is based on numerical convergence point search. They commonly have equal dimension number ($n = 4$) and has small differences. In this part, we use only range distribution and the output is Figure 3.29. Left one (X) shows the vector map including calculated homing direction at each point with blue arrows in $n = 4$ and the output is almost same. Right one shows the changes of homing accuracy according to dimension number from 1 to 10.



(a) (b) (c) (d)

Figure 3.27: Robot experiment result with estimated distances. Using perimeter moment model : (a) shows the case without color, (b) with estimated distances. Using area integration moment model : (c) without color, (d) with estimated distances. The red box is home located at (500,500) and the arrows show the homing direction at each point.

3.2.3.4 Comparison of overall methods

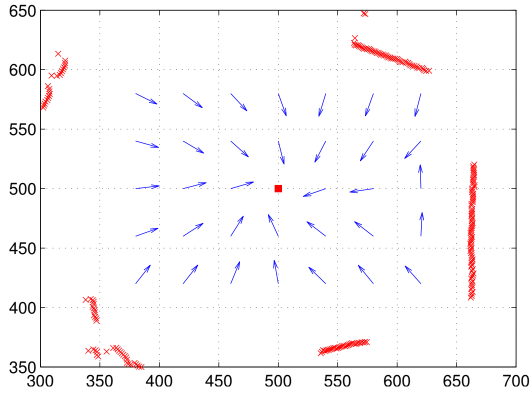
In prior parts, we test our methods using real robot experiment. In this part, we analyse our outputs compared with other algorithms called DID. Based on these results, we check overall performances through various conditions. The outputs are shown in Figure 3.33 and Table 5.1.

3.2.3.5 Applying to other environment

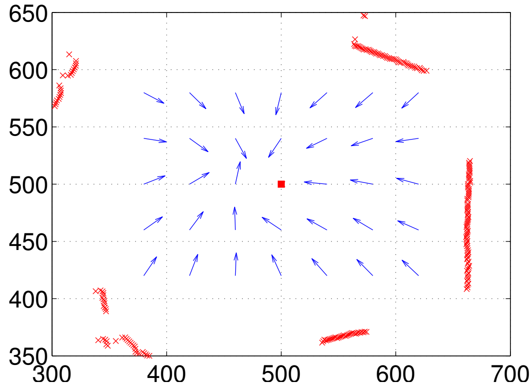
Next, we apply our model into other dataset. In this part, we use Vardy's open dataset including several panoramic images which is broadly used in this field. It is the one

Table 3.5: angular errors of different methods

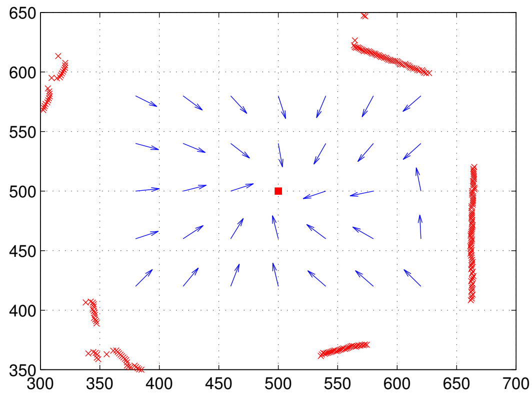
		error μ ($\pm\sigma$)	N	$0 \leq \varepsilon_\theta < 45°$	$45 \leq \varepsilon_\theta < 90°$	$90 \leq \varepsilon_\theta < 180°$
With reference	I	21.6 (±11.9)	34	97.1%	2.9%	0%
	II	19.4 (±11.6)	34	97.1%	2.9%	0%
	III	13.4 (±8.8)	34	100%	0%	0%
	IV	28.3 (±18.7)	34	91.2%	8.8%	0%
	V	21.9 (±13.6)	34	100%	0%	0%
	VI	10.4 (±6.2)	34	100%	0%	0%
	VII	9.3 (±5.7)	34	100%	0%	0%
	$VIII$	8.2 (±4.4)	34	100%	0%	0%
	IX	7.6 (±3.6)	34	100%	0%	0%
	X	8.8 (±7.1)	34	100%	0%	0%
Without reference	I_v	23 (±15)	34	94.1%	5.9%	0%
	VII_v	17.6 (±18.6)	34	94.1%	5.9%	0%
	VII_r	18.1 (±16.2)	34	91.2%	8.8%	0%
	IX_v	14.6 (±16.2)	34	94.1%	5.9%	0%
	IX_r	13.1 (±7.3)	34	100%	0%	0%



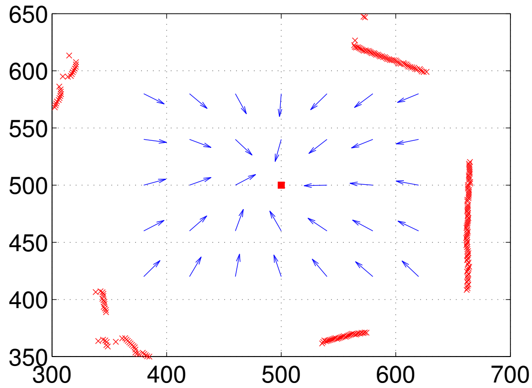
(a)



(b)



(c)



(d)

Figure 3.28: Robot experiment result using both measured distances and color intensities with two alignment algorithms. Aligned by visual compass : (a) shows the case with perimeter, (b) with area integration model. Aligned by rearrangement algorithm : (c) shows the case with perimeter and (d) with area integration model. The red box is home located at (500,500) and the arrows show the homing direction at each point.

called 'A1originalHh' used in Andrew Vardy's experiment (Möller et al., 2007). They are from 'http://www.ti.unibielefeld.de/html/research/avardy/'. Area is 2.7m x 4.3m and there are 170 points with 10cm intervals.

Figure 3.30 is the output and we can check the homing convergence in cases whose n variable over 2. In this part, we use estimated distances, not measured one. Then the output cannot be directly equal to the one of measured one but overall patterns are similar.

Figure 3.31 shows the outputs with aligned condition. These models use estimated distances. First one (II^P) uses generalized moment model with perimeter without color

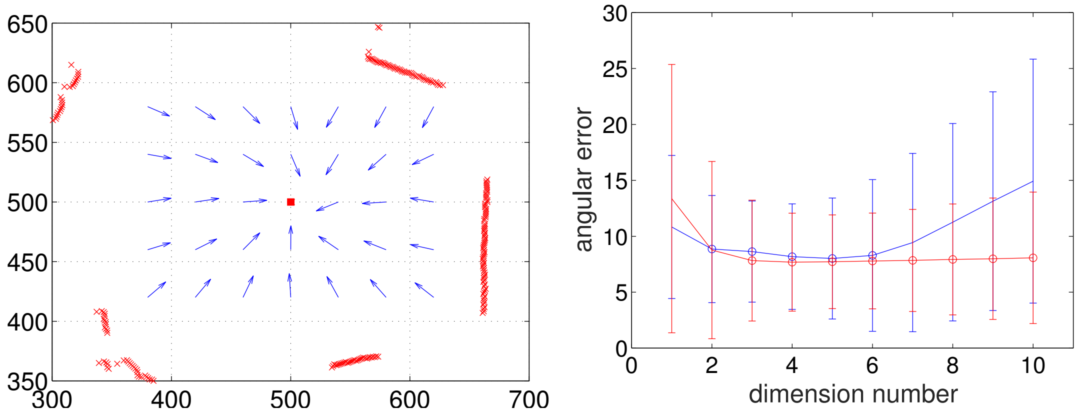


Figure 3.29: Blue line shows the output of multidimension cases using derivative and red one for numerical method. Each circle means statistically minimum positions in each distribution.

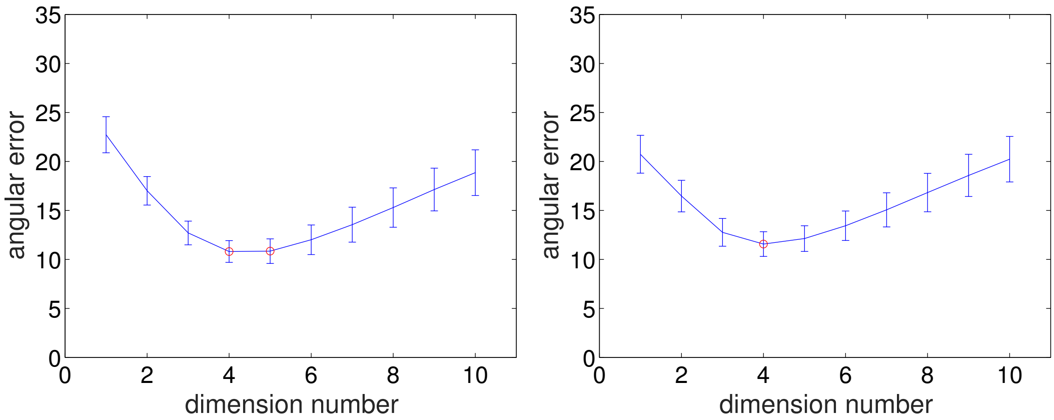
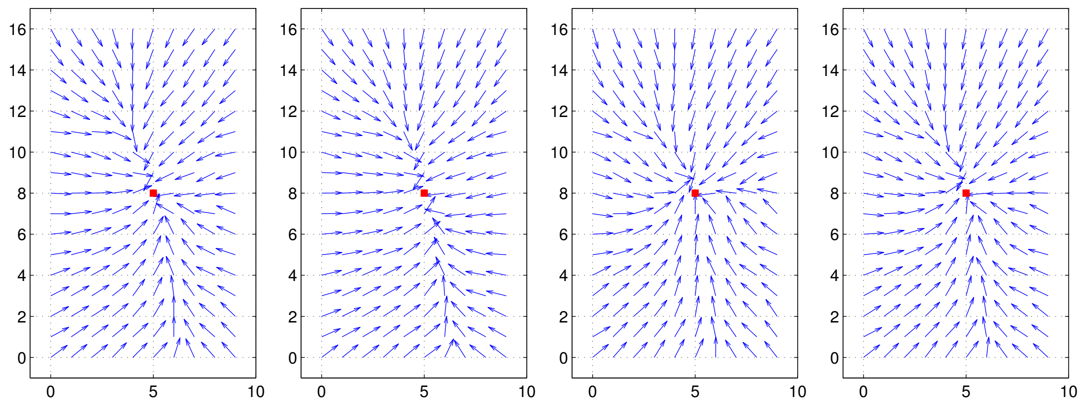


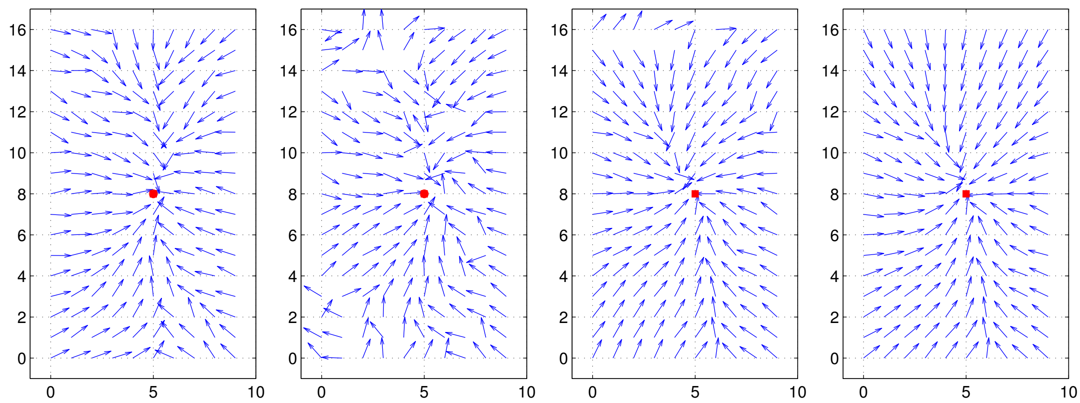
Figure 3.30: Performance graph for homing accuracy according to changing dimension number (n) of generalized moment function. Left one is without color information and right one is with both color and range. Red dots means the minimum point and statistically equal point.

information and second one (III) with both range and color intensities. Third one (IV) uses generalized moment model with area integration method without color and fourth one (V) with color. Figure 3.32 shows the outputs with alignment algorithms. These models also use estimated distances. First one (III_v) uses generalized moment model with perimeter and second one (V_v) uses area integration model with visual compass. Third one (III_r) uses generalized moment model with perimeter method and fourth one (V_r) uses area integration model with rearrangement algorithm. The outputs are also shown in Table 5.2.



(a) (b) (c) (d)

Figure 3.31: Robot experiment result with estimated distances. Using perimeter moment model : (a) shows the case without color, (b) with estimated distances. Using area integration moment model : (c) without color, (d) with estimated distances. The red box is home located at (500,500) and the arrows show the homing direction at each point.



(a) (b) (c) (d)

Figure 3.32: Robot experiment using Vardy's image set result using only color intensities with horizontal line. (a) is the output using DID with pre-aligned images. (b) uses DID aligned by visual compass. (c) uses area integration model aligned by visual compass and (d) aligned by rearrangement. The red box is home located at (5,8) and the arrows show the homing direction at each point.

3.2.4 Experiment with Occlusion

First, we focus on the issues of snapshot-based models. Among these issues, the occlusion problems under dynamic circumstances are challenging objective for our model. Therefore, we have added some experiments and assessments to evaluate the model

Table 3.6: angular errors of different methods in Vardy's set

		error μ ($\pm\sigma$)	N	$0 \leq \varepsilon_\theta < 45°$	$45 \leq \varepsilon_\theta < 90°$	$90 \leq \varepsilon_\theta < 180°$
With reference	I	23.31 ($\pm$13.66)	169	97.1%	2.9%	0%
	II	17.01 ($\pm$9.64)	169	100%	0%	0%
	III	20.73 ($\pm$12.78)	169	97.1%	2.9%	0%
	IV	10.81 ($\pm$7.37)	169	100%	0%	0%
	V	12.77 ($\pm$9.4)	169	100%	0%	0%
Without reference	I_v	33.30 ($\pm$34.54)	169	79.88%	11.83%	8.29%
	III_v	26.96 ($\pm$20.76)	169	88.76%	8.28%	2.96%
	III_r	20.74 ($\pm$12.96)	169	97.1%	2.9%	0%
	V_v	17.55 ($\pm$20.76)	169	96.5%	0%	3.5%
	V_r	12.97 ($\pm$9.42)	169	100%	0%	0%

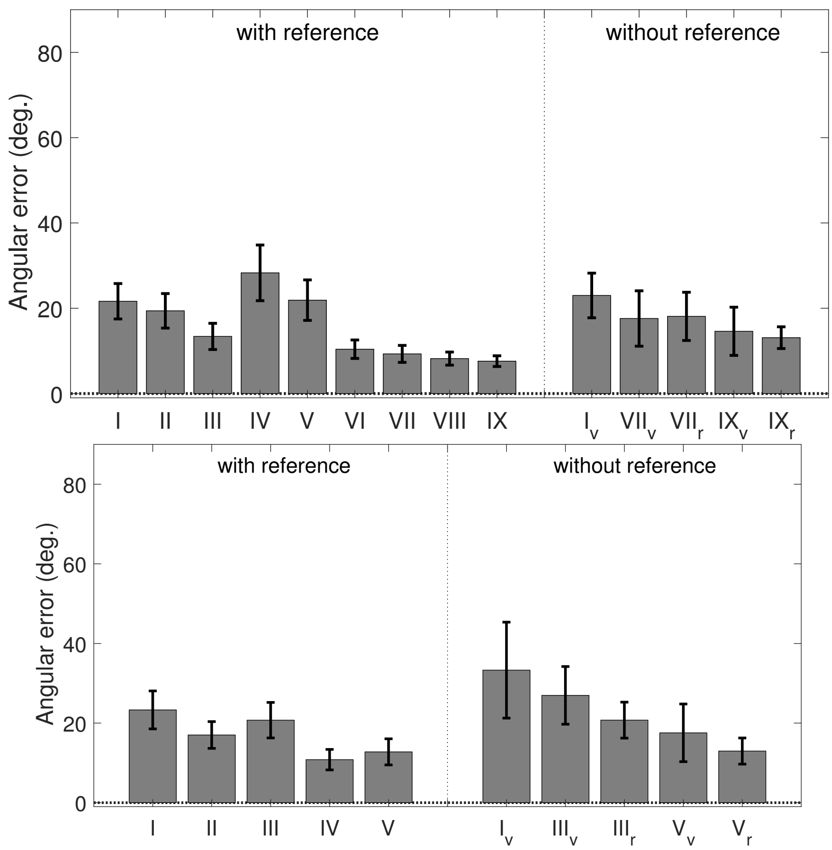


Figure 3.33: Performance graph between various models and angular errors. First one is with our dataset and second one with Vardy set.

under more difficult circumstances involving occlusion issues and dynamic conditions.

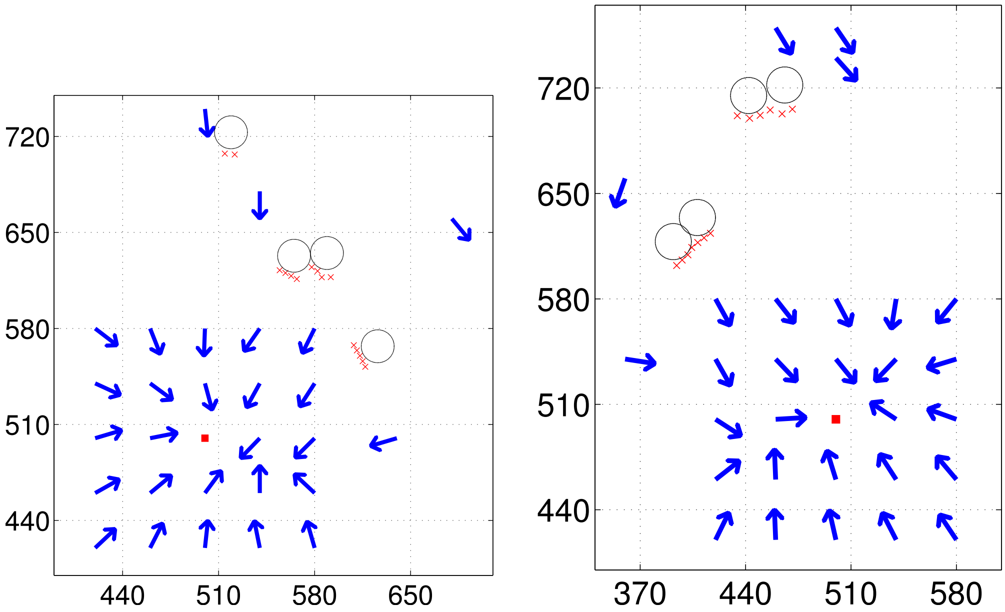


Figure 3.34: Occlusion test using moment model ($n = 2$). In the same field of 'ourlab' environment, we placed four trashcans together making a convex hull and calculated the homing direction at various locations including inside, around, and outside of the convex hull. The red rectangle at (500,500) represents the home position and the blue arrows show the calculated homing direction.

Our model aims to the find partial solutions for occlusion problems. This occlusion can be generated for many reasons and can cause large errors in homing problem of snapshot-based methods. Therefore, we have designed a new experiment that includes occlusion outside the convex hull. The convex hull is distinguished using the boundary connecting all the surrounding landmarks. Therefore, measurements outside of the convex hull will have occlusion. In contrast to past experiments, we calculate the homing direction at these points. Figure 3.34 shows the output. There are two environments having different object arrangement. The image on the left has 24 points on the inside of the convex hull, 3 points around the convex hull, and 1 point on the outside of the convex hull. The image on the right also has 24 points on the inside of the convex hull, 4 points around the convex hull and 1 point on the outside of the convex hull. In this part, we see that the surround ones show homing directions with smaller homing errors under 45°. However, two cases on the outside show poor results. Therefore, we can see the necessity of an additional home image at the boundary of the convex hull using the limit of locality in snapshot model to apply in wide range navigation.

Next, we perform another experiment under dynamic circumstances by artificially

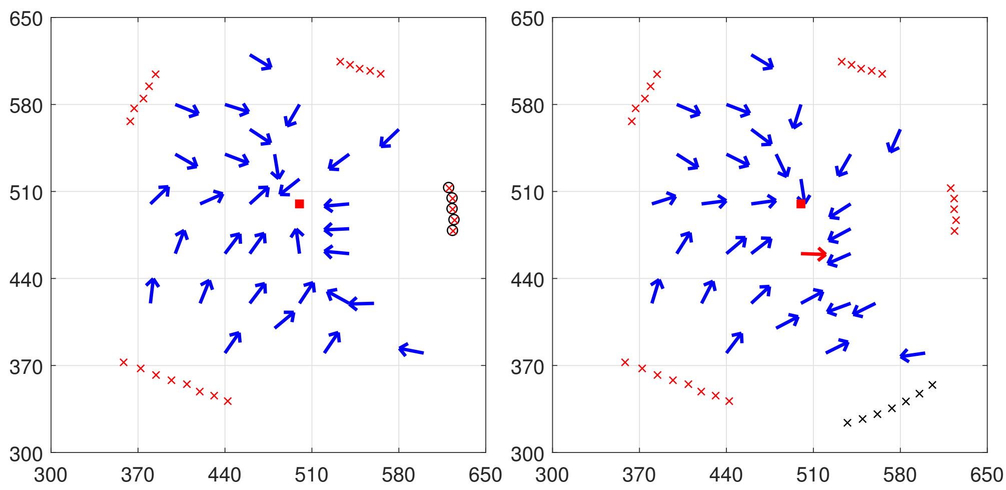


Figure 3.35: Homing test with dynamic environments. On the left, we see some objects marked by black circles to far position (similar with the wall position) and right one moves some parts marked by black x-marks to near position (similar with other object position). The red box is the home position and red x-marks are used for measured landmarks. The blue arrows are used to calculate the homing direction and have an error smaller than 45° and red are not.

changing measurements. We control the home measurement by moving some portion. Figure 3.35 shows the output. Left one having small manipulation shows successful in overall locations but right one shows some fails in manipulated directions. Therefore, we try to check the effect of the manipulation ratio on the homing direction by changing this area. The output is Figure 3.36. Left one moves them to far points and right one moves to near points with increasing the area. Increasing the size of this area lowers the overall output . With this, we can verify that our model can operate in cases having less than 100 degree manipulations. The performance worsens significantly beyond 100 degree manipulations.

3.3 Summary of Chapter 3

In this chapter, we suggest a series of new navigation methods using various moment models. Specifically, our first method can figure out homing direction by using the unique convergence point from reference map. We expand this moment function to modified versions using various features, various dimension numbers and we check each operation. In this part, we check that there is the unique convergence of first mo-

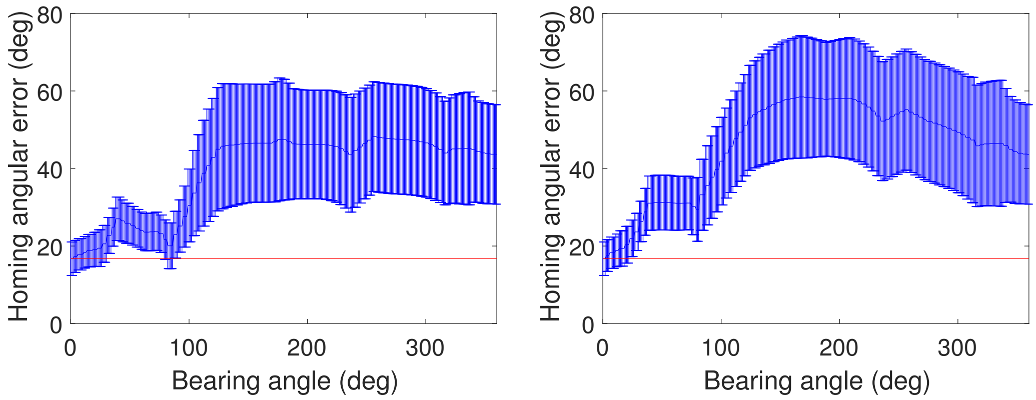


Figure 3.36: Performances of homing tests based on the area of the dynamic region. These are the changes in the size of the manipulation area in each case. The x-axis shows the size of area (in degree) and the y-axis shows angular error. Each error bar shows both mean and length of confidence interval (95 percent with assumption of t-distribution).

ment model in homing problem, and changing dimension number has various effects, which is started from inter-filled case. First, if this number is over 1, we prove that there must be unique minimum point. Second, if this number is negative, it can operate like obstacle avoidance. As our model does not use any object extraction, the output of landmark vector summation to omni-directional measurement can be different from the original purpose. Therefore we try to fill the space with landmark information by area integration. We have focus on the models using higher dimension and we find remarkable features from our situation. Then we can figure out that the one using higher dimension number ($n > 2$) in moment potential is more appropriate to the original purpose than normal one ($n = 2$).

Chapter 4

Visual Navigation Based on Visual Masks

In this chapter, we apply a concept of visual masks to our new navigation model based on snapshot model of holistic method. Through this part, we try to find the effective model of visual receptors in homing problem. For instance, in biological vision system, there are numerous receptors in retina. Both cone and rod cells directly accept visual cues like filter. Bipolar cells receive synaptic inputs from either rods or cones, or both of them (Paik and Ringach, 2011), to find shape information and these are considered as the basic unit for visual perception. Many insects with bipolar cells demonstrate excellent homing ability even though they have poor resolution. Then, first, we partially imitate the structure of bipolar cell by using the various types of haar-like feature, and apply generated multilayer masks on horizontal line into homing navigation. Second, we made a new model by adding randomness in mask generation process. We argue that the difference of matching score for each mask can indirectly estimate the distribution of landmarks in the environment, and it can ultimately determine the homing direction. Also, using additional reference images like learning phase of insect, we try to find a robust homing method based on the image differences. We demonstrate the effectiveness of the method in several environments. The one using visual masks is under the preparation for submission (Lee and Kim, 2017d) and the other one using randomly generated haar-like masks is under review (Lee and Kim, 2017e). Detail procedures of our model is depicted in next part.

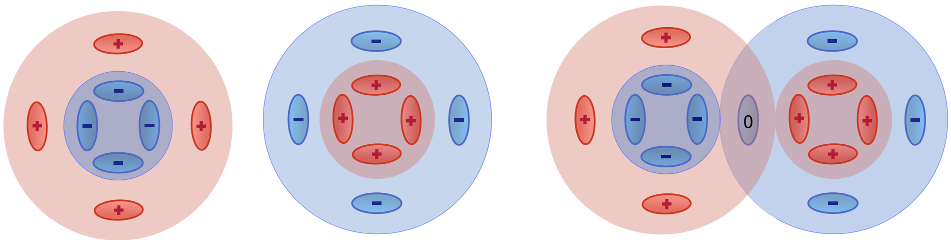


Figure 4.1: Bipolar cell model. The first is an ON cell and the second an OFF cell. They are composed of two regions: the blue region indicates increased activity, and the red region indicates decreased activity. The shown combinations operate either an ON or OFF switch. The ON cell fires when the surrounding pixels are darker than the central pixel, whereas the OFF cell fires in the opposite case. The bipolar cell is a combination of both cells, shown at the right of the graph.

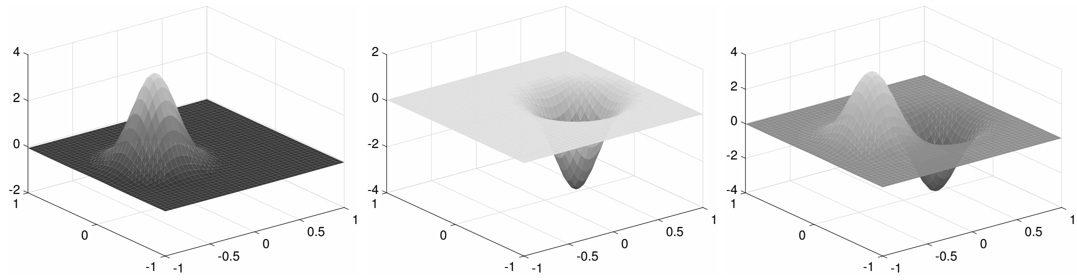
4.1 Visual Mask Modelling

The bipolar cell model, which is a fundamental concept of the proposed method, considers both ON and OFF cells. In fact, the function of these cells located in the retina are basic concepts for image processing in neural networks. The ON and OFF cells are excited by brightness and darkness, respectively, and are composed of positive and negative regions, as illustrated in Figure 4.1. In the cells, one region has a larger area than the other, and the regions are represented as concentric circles. The function of the cells is determined by the combination (i.e., difference) of activity in the two regions.

$$Z_{on} = \frac{e^{-\frac{(X-\mu_{X_{on}})^2+(Y-\mu_{Y_{on}})^2}{2\sigma_{on}^2}}}{2\pi\sigma_{on}^2}, Z_{off} = -\frac{e^{-\frac{(X-\mu_{X_{off}})^2+(Y-\mu_{Y_{off}})^2}{2\sigma_{off}^2}}}{2\pi\sigma_{off}^2},$$

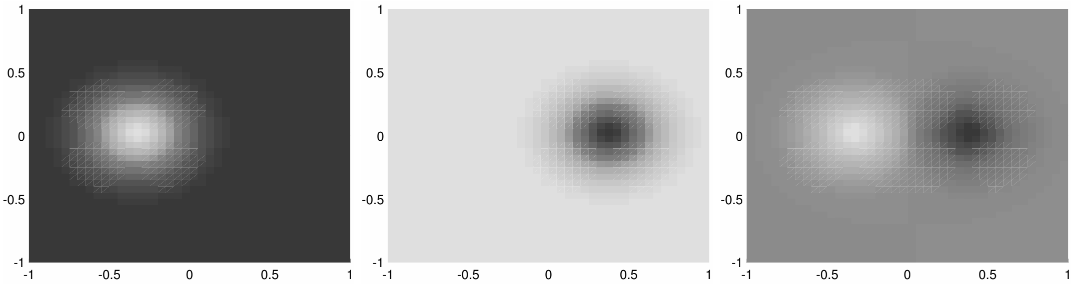
where Z_{on} and Z_{off} represent the firing rate of the ON and OFF cells, respectively. σ_{on} and σ_{off} are standard deviation of distribution of each cell. μ_X and μ_Y are center location of each cell. Hence, these cells exhibit exponential firing rates and require location and deviation information. However, ON and OFF cells exhibit an opposite behavior, as the ON cell reacts to brightness, whereas the OFF cell reacts to darkness in the input image.

The bipolar cell can be modelled by summing the firing rates of the two types of cells:



(a) (b) (c)

Figure 4.2: Bipolar cell activity of the (a) ON cell, (b) OFF cell, and (c) bipolar cell that combines the activity of both cells.



(a) (b) (c)

Figure 4.3: Structural form of the cells used in the proposed method. Basic (a) ON and (b) OFF cells, and (c) bipolar cell with its activity shown along the x-axis direction.

$$Z_{bipolar}(X) = Z_{on}(X - \lambda, \mu_{Y_{on}}, \sigma_{on}) + Z_{off}(X + \lambda, \mu_{Y_{off}}, \sigma_{off})$$

$$Z_{bipolar} = Z_{off} + Z_{on}.$$

where λ is the distance between two cells and σ represents the size variation of the corresponding cells. $Z_{bipolar}$ is approximated simulation output of bipolar cell. Hence, this model can be obtained from the sum of the activity of neighboring ON and OFF cells. This model can achieve the operation of a line detector. In addition, we denote the smallest bipolar cell as being level 1. Figure 4.2 shows the outputs of the bipolar cell model. The ON cell firing shows a peak at the center, whereas the OFF cell firing shows a negative peak at the center. The z axis corresponds to the activation level of the cells according to location, and it resembles a Haar-like feature. The figure also shows that the bipolar cell combines the activity of the two cells. Figure 4.3 shows the top view of the activity of these cells. Two peaks can be seen in the bipolar cell model, which can operate like an edge detector. This set of bipolar cells constitutes a basic component of the proposed method.

Based on bipolar cell reaction, we approximate the operation by using simple Haar-like mask called visual mask. Haar-like mask has two region and find its matching score with input image. If there is a snapshot image I available, the matching score $S(I,M)$ for a given Haar-like mask M is calculated as

$$S(I,M) = |\sum_{i\in M+} I_i/m_1 - \sum_{i\in M-} I_i/m_2| \tag{4.1}$$

where I_i indicates the i-th pixel of the image and $m_1 = \sum_{i\in M+} 1, \quad m_2 = \sum_{i\in M-} 1$ are the size of white and black regions, respectively. The score calculates the difference of the mean values of pixel intensities of white and black regions. In this part, we try to approximate the on-cell with white region and off-cell with black region.

4.1.1 Navigation Using Visual Mask

4.1.1.1 Multi-level visual mask set generation

Using simple sets of visual masks can be inadequate when considering the complexity of a real visual environment. Hence, we define larger sets having more complex mask structures than the masks of level 1, corresponding to levels 2 and 3 (we consider level 3 as the maximum level) in order to obtain size-invariant characteristics.

To define the additional levels, we compose various masks along the horizontal line. First, we compose visual masks along the x axis that contain one ON and one OFF region of different sizes. This composition allows vertical edge detection, which is important for visual navigation using omnidirectional views. In addition, agent motion can change the perceived size of visual cues. To avoid such variations, we use the concept of a multiscale image pyramid. Therefore, each mask set is composed of equidistant visual masks along the x axis, which have different sizes according to the above formula. Consequently, unlike a common visual mask, the proposed variation includes scale parameter.

Next, we compose mask sets at four different levels. The above formula shows the generation of variable-size visual masks to perform edge detection. We use three levels of visual masks and define one more type of mask set called level 0. This level also contains bioinspired elements, which correspond to cone cells that react to a certain

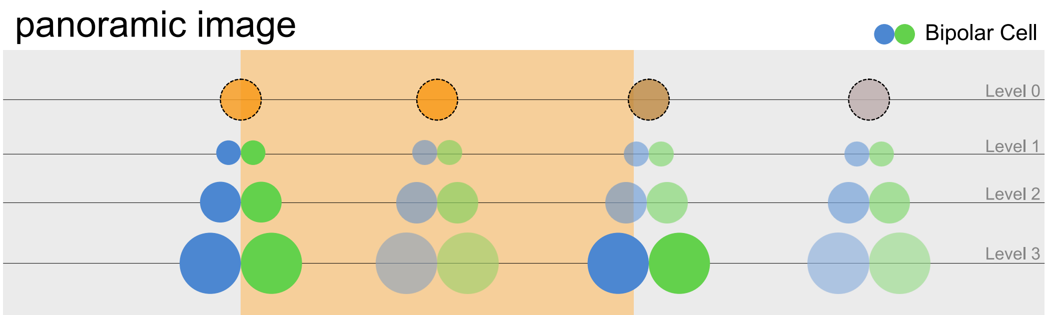


Figure 4.4: Multiscale mask sets. The background represents an omnidirectional image that includes a yellow object. Four sets of equidistant masks are depicted. Level 0 behaves like a cone cell for color acquisition. Levels 1 to 3 contain visual masks for vertical edge detection at various scales. The mask brightness at each level determines the output.

visible light bandwidth, and it acquires the color of objects by combining their signals. Therefore, we use this level for color detection in the image filtering process.

From the abovementioned definitions, we can construct mask sets that describe four levels, from 0 to 3. Then, we repeatedly generate them along the horizontal line to determine a group of 720 equidistant masks per level. The generated masks are used as windows to evaluate input images. For instance, Figure 4.4 illustrates a general situation, which shows a background image including a yellow landmark and the four levels. The first column of masks perfectly acquires the edge, and hence their output is bright at all the levels. The second and third columns show a low brightness, as they are not aligned with the landmark edge. The third column output is bright only at level 3, as it is slightly at the right of the edge. The columns are located along the horizontal line to have a resolution of 0.5 $^\circ$ when evaluating an input image.

4.1.1.2 Applying generated mask to input visual cue

After generating the outputs from the mask levels, we apply them in an image processing stage. We use the visual data from the surroundings of an omnidirectional camera. A raw picture obtained from the camera which cannot be directly used to perform the intended navigation task.

To process this raw picture, we first set its center and the region to use for navigation. In the figure, the red dot represents the manually set picture center. In this part, we do not use any pre-processing technique such as histogram equalization, because the

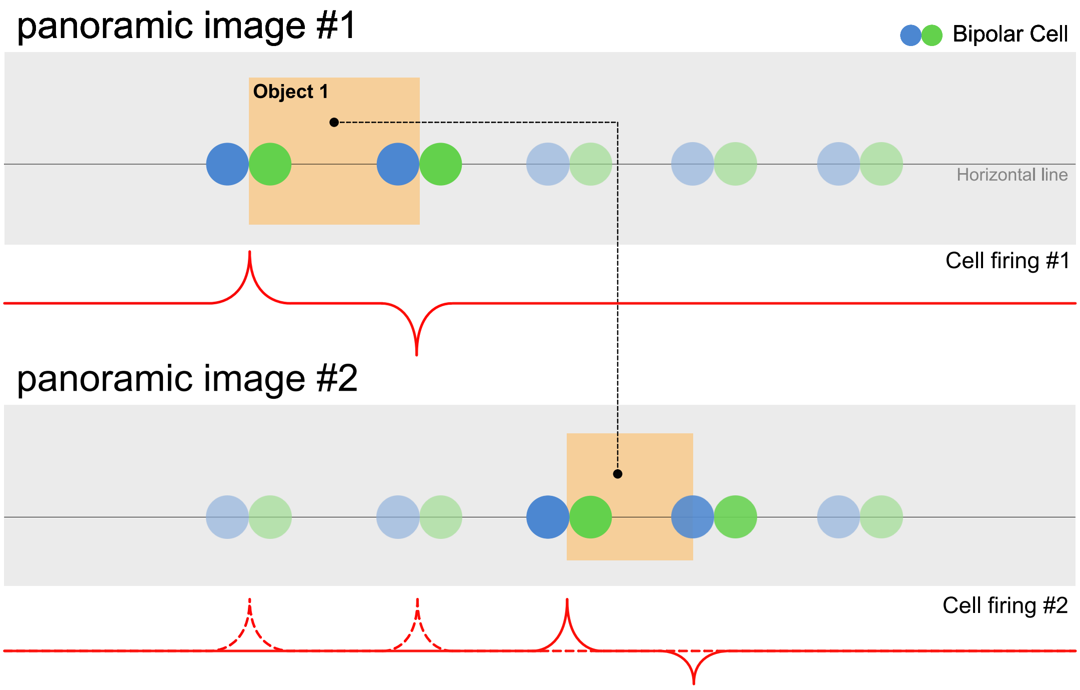


Figure 4.5: Visual mask firing. The masks are placed along the horizontal line. The background and yellow rectangle represent the image plane and an object, respectively. Each visual mask fire according to the location of the object edge, as illustrated in the red curves below each image. It can be seen that the masks operate as vertical edge detectors, and the firing rates vary as the object moves.

used dataset has observation stability that makes such additional processes unnecessary. However, if the proposed method is applied, for instance, to images that present illumination variations, additional processing might be required. To perform navigation, we use two of the abovementioned images, one corresponding to the home and the other to the current position, in order to apply the snapshot model.

4.1.1.3 Visual mask set activity

The above sections present the generation of mask sets and the process to determine image alignment. The next step is to combine these aspects. As mentioned above, we use the generated mask sets as windows to evaluate the surroundings and detect visual cues as the image changes.

First, we obtain the activity of each mask in the corresponding set. Each mask fires according to the section of the image that it covers, thus returning the corresponding activity sum as output. To illustrate this activity, Figure 4.5 shows visual masks that are

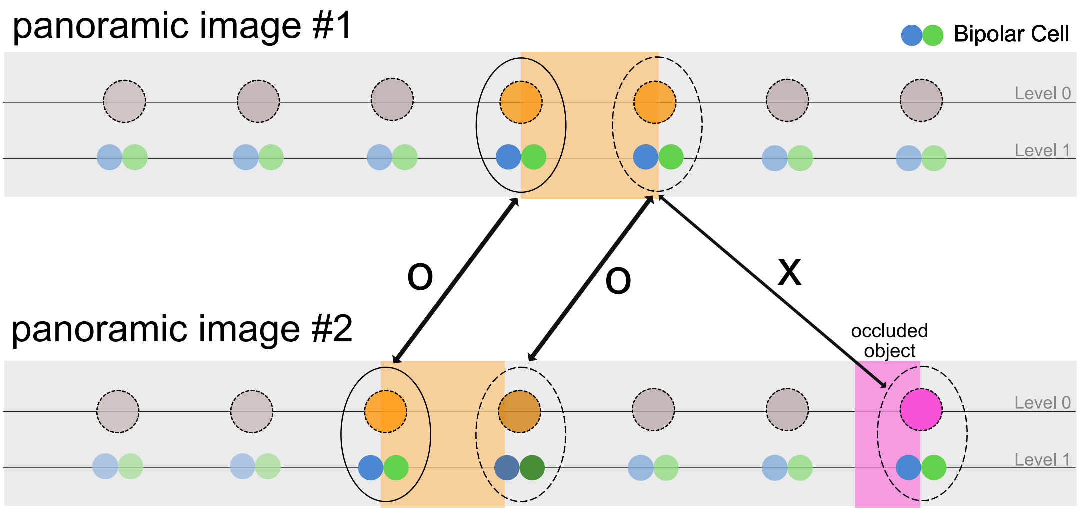


Figure 4.6: Matching algorithm operation. Level 0 retrieves color information along the horizontal line. The yellow and pink rectangles represent objects with different color information. There are seven mask columns, and two levels are illustrated (we omitted levels 2 and 3 for simplicity). Mask brightness represents its activity in the graph.

arranged along the horizontal line and evaluate two images. The corresponding mask firing is shown as the red curves below the images. When a landmark object moves, a different pattern of mask firing is obtained. Then, we collect all the mask outputs and generate the output for the i-th level mask as $S_{i\theta} = M_i(\theta) * I$ according to direction θ. Nevertheless, using only visual mask activity is not be sufficient for homing navigation. Therefore, we add a matching and filtering algorithm for the level outputs.

4.1.1.4 Matching and filtering mask set output

This stage aims to increase the robustness of the mask outputs between two images. First, we retrieve color information along the horizontal line by using the masks at level 0. Next, edge information is determined from masks at levels 1 to 3. Here, if the outputs of masks at these levels are below a threshold, the corresponding mask is inhibited. These steps are affected by the input image and consequently by the landmark location, and hence the output corresponding to home and another position images will be notably different.

Next, we calculate the matching scores among mask activity, where the noninhibited masks from two images are compared to each other in order to obtain the matching scores at each level. The matching score is the inverse of the difference between out-

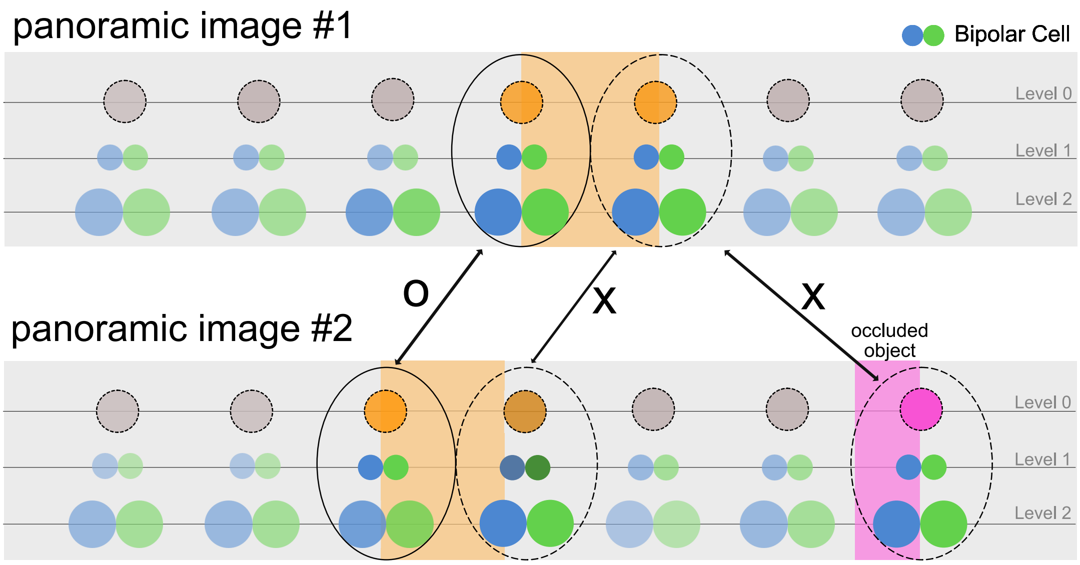


Figure 4.7: Filtering algorithm operation. Unlike Figure 4.5, high mask activity is exhibited in two columns at the right of the first active column. To solve ambiguity, the two connections marked with X are removed.

puts. Then, similar images result in a high score, as illustrated in Figure 4.6, which shows three cases. The activity of the first group of circled masks clearly matches. In addition, the activity of the second and third group of circled masks match at level 1. However, masks at level 0 exhibit a matching divergence. At this stage, we consider the connections between active masks as having weights that are equal to the matching score. Hence, similar matches will generate even higher scores.

Finally, we filter out ambiguous cases. This process is designed to remove misleading matches, as shown in Figure 4.7. The second and third matching masks show similar activity and are connected with corresponding weights. However, a single connection should be present, and to select either one can generate an incorrect result. Therefore, we delete both connections to remove ambiguity. In addition, we increase the weight (score) value where no ambiguity appears, and select a specific amount (e.g. 25%) of matches with the highest rates to be considered in the navigation process, thus increasing robustness.

After the matching and filtering process, we determine landmark vectors that represent location information according to the obtained mask activity outputs. The obtained score values result from the matching weights between masks according to location and size. We define each landmark vector by considering the angular position as the angle

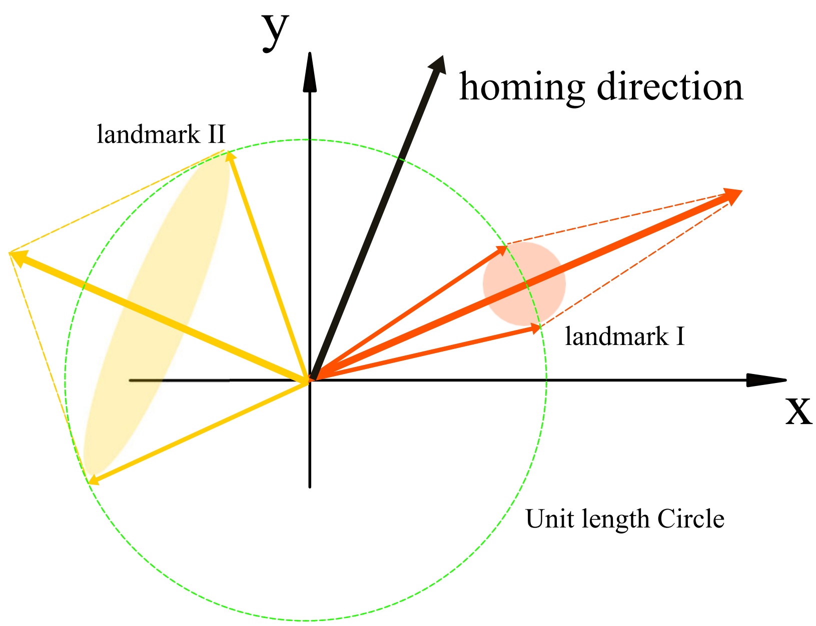


Figure 4.8: Landmark vector extraction corresponding to Figure 4.5. The vectors are constructed considering matching weights and angular locations of the landmarks obtained from mask activity. Then orange and yellow arrows represent the mask activity (i.e., edge detection) obtained from the first and second images, respectively. The thick arrows represent the sum landmark vectors. The homing direction is determined by subtracting these representative vectors.

of the landmark with respect to the center of each mask column, and the magnitude as the obtained weight value. Then, we generate landmark vectors according to the number of obtained matches in both images. Finally, we combine the landmark vectors from the home and current position images.

$$(S_{H,i}, S_{C,i}) = (M_i(\theta_i) * I_{Home}, M_i(\theta_i) * I_{Current}) \tag{4.2}$$

where $S_{H,i}$ is score of i-th mask in home image (I_{Home}) and $S_{C,i}$ in current image ($I_{Current}$). Hence, the sum of each of the two landmark vector sets from the images can be considered as representative vectors for each location, according to the snapshot model. Therefore, we can find the homing direction by subtracting their values:

$$\overrightarrow{H} = \sum_{i=1}^{N} \overrightarrow{LV_H} - \sum_{i=1}^{N} \overrightarrow{LV_C} = \sum_{i=1}^{N} (S_{H,i}cos(\theta_i) - S_{C,i}cos(\theta_i), S_{H,i}sin(\theta_i) - S_{C,i}sin(\theta_i)) \tag{4.3}$$

Figure 4.8 illustrates the landmark vectors and homing direction for the images in

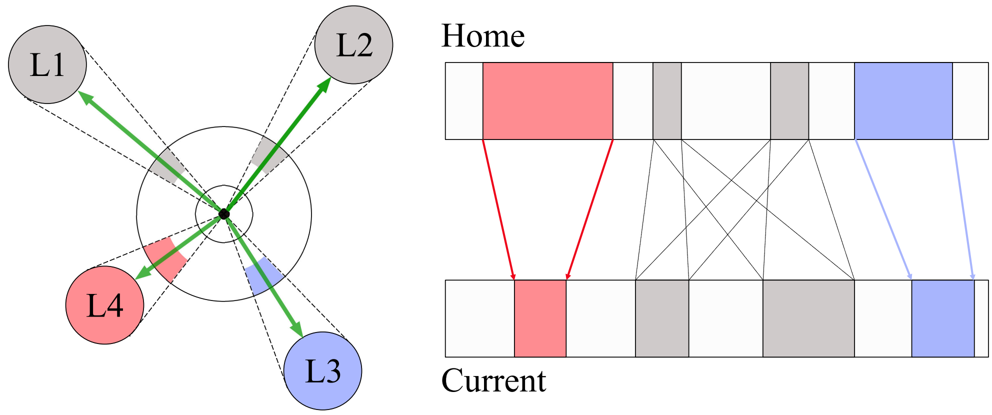


Figure 4.9: Landmark vector types. Top view showing the arrangement of landmarks that are represented as circles (left). The corresponding omnidirectional image is delimited by the concentric circles around the central black dot. The green arrows represent the original landmark vectors. Without clustering, the pixel-wise landmark can be either the red (using the estimated distance based on the ground line) or blue regions (using only angular position of pixel landmark).

Figure 4.5, and we can visually compare the firing rates to the obtained vectors. It can be seen that landmark vectors correspond to each activity peak, where vectors are indicated by the thin arrows and their sums correspond to the thick arrows in Figure 4.8. The homing direction is represented by the thick black arrow in the figure.

The homing direction allows to determine the motion necessary to reach home from any other position. Given that this method does not use any distance sensor, the calculated direction must be accurate. Moreover, we verify the homing direction at each iteration by comparing the current and home locations. In the next section, we prove the global convergence of the proposed method. Therefore, we guarantee that the agent using the proposed navigation method reaches the home position.

4.1.1.5 Global convergence to the home position

We now describe the method in mathematical terms. First, we model the environment from where images are acquired. We assume that each landmark is a cylindrical object without occlusion in the experiment area. The cylinder is a shape commonly used in navigation, because it does not show pose variation along the horizontal direction, thus simplifying its model. In addition, many types of objects can be approximated by cylinders, and including complex and varied shapes for the landmarks would consid-

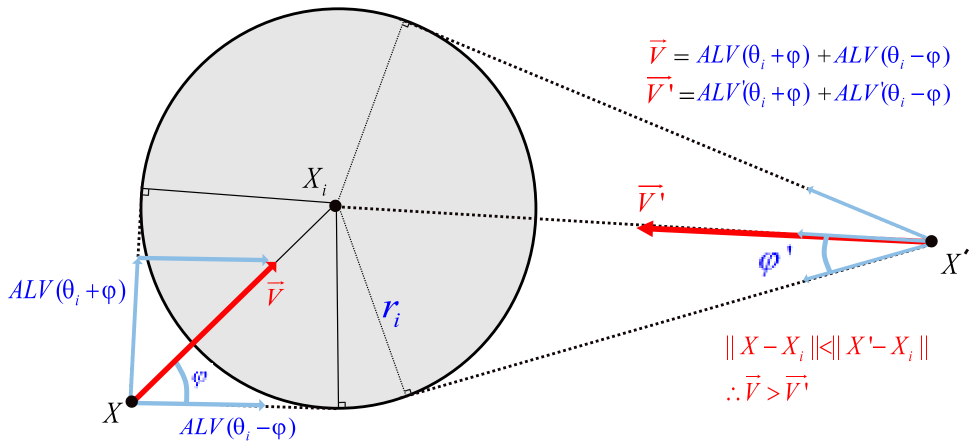


Figure 4.10: Principle and concept of pixel-wise landmarks. The graph shows a top view of the landmark cylindrical object (gray circle). X_i is the center of the cylinder with radius r_i. Two observations are considered at points X and X'. Equal length vectors called ALVs point to the object edges, as represented by the blue arrows. Their sum at each observation point results in the red arrows, which are not geometrically equal to the real landmark vector.

erably increase the complexity of the proposed navigation method. This would also contradict the simplicity we aim to attain, such as that of the visual mask structure.

Figure 4.9 illustrates the basic principle of the proposed navigation method. The environment is considered as flat, and both the floor and background are assumed to be homogeneous white surfaces. In addition, we delimit the floor by using a horizontal line. The figure shows four landmarks (from L1 to L4) that have a cylindrical shape and different colors. The input image corresponds to the panoramic projection of the circular image shown at the left, which resembles the honeybee vision. The green arrows represent landmark vectors containing both angular and distance information, which can be retrieved by using a range sensor and a clustering algorithm. Given that our experimental robot does not include either of these, we use robust the matching edges obtained from the visual mask sets and processes that constitute the proposed method.

The proposed method emulates visual masks that exhibit their maximum activity when acquiring vertical edges and uses only angular information of these edges. However, these mask landmark vectors (CLV) are different to those obtained from sensing and clustering techniques. Hence our assumption of a cylindrical shape for landmark objects.

Similar to the ALV method (Lambrinos et al., 2000), overall summation of unit landmark vector can be expressed by

$$\overrightarrow{O} = \sum_{\theta} \overrightarrow{V_{\theta}} = \sum_{i} w_i \overrightarrow{u}(\theta_i). \tag{4.4}$$

where we use the CLV summation ($\overrightarrow{O}$) to represent an image location observed from point X. As vector summation is commutative, the CLVs of one objects can be defined by a weighted sum of unit vectors $\overrightarrow{u}(\theta_i)$.

$$\begin{aligned} \Lambda &= ||\overrightarrow{V}|| = 2(1-\tfrac{R_i^2}{r_i^2}), \\ \overrightarrow{V} &= \overrightarrow{u}(\theta_i+\varphi) + \overrightarrow{u}(\theta_i-\varphi) = 2(1-\tfrac{R_i^2}{r_i^2})\overrightarrow{u}(\theta_i). \end{aligned} \tag{4.5}$$

where $r_i(X)$ be the distance between observation position X and landmark location X_i, and the radius of the landmark object be R_i. Figure 4.10 shows that the length of the vectors pointing to the object does not correspond to the distance, but it is proportional to the shape (see details in Appendix). Therefore, we adjust the model for a multi-object environment by

$$\nabla M = \sum_i 2(1-\frac{R_i^2}{r_{i0}^2})[X_0,Y_0] - 2(1-\frac{R_i^2}{r_i^2})[X,Y]. \tag{4.6}$$

where ∇M is calculated homing vector form potential field of this model. r_{i0} is distance between i-th object and home location (X_0,Y_0) and r_i with current location (X,Y). In addition, we verify that the three conditions for global convergence to the home position are satisfied:

$$\begin{aligned} &\exists i,j(i \neq j) : (y_i-y)(x_j-x)-(x_i-x)(y_j-y) \neq 0 \\ &M_{xx}(X) > 0, D(X) > 0, \nabla M(X_0) = 0. \end{aligned} \tag{4.7}$$

where (x_i,y_i) is i-th location of landmark and (x_j,y_j) is j-th location of landmark. M_{xx} is second derivative of x in field potential (M), $\nabla M(X_0)$ for gradient of potential which is equal to homing vector at home position and D is determinant. Until the adjustment of the landmark object does not satisfy the conditions above, the proposed method provides the direction to achieve home through the homing vector (see details in Appendix).

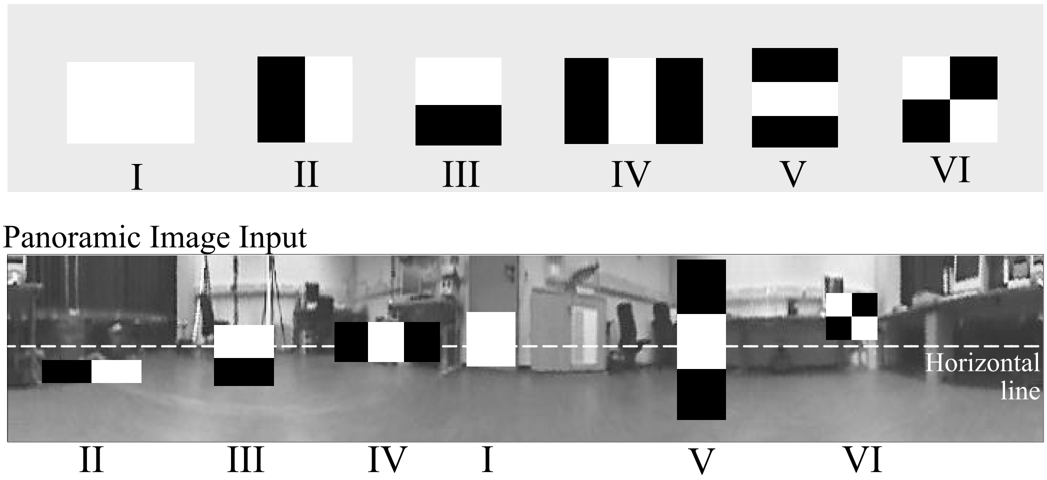


Figure 4.11: Examples of Haar-like masks and their matching in the image; six types of masks with one or more rectangles. Type I has one rectangle, Type II and III two regions, Type IV and V three regions, and Type VI four regions; lower picture shows a panoramic image and matching of those Haar-like masks.

4.1.2 Navigation Using Landmark Model with Various Haar-Like Masks

Based on prior modelling, we try to apply randomness in mask generation like bio-inspired retina. In our approach, 5000 Haar-like features are randomly generated and applied to the two snapshot images. The feasures have random sizes, patterns and locations in the image. Figure 4.11 shows examples of Haar-like features and their matching in the snapshot. The matching score can be calculated as how closely a Haar-like feasure match a patch in the snapshot image. Each Haar-like feature can be a landmark candidate at its angular position in the snapshot. We assume that the change of the snapshot image depending on the displacement of an observer's position can be reflected on a set of Haar-like mask scores. The matching score of each Haar-like feature relies on the pattern matching. If the feature has similar pattern to a patch in the snapshot image, the score becomes high. The difference of a Haar-like feature score measured in a pair of snapshot images can provide a sign of moving direction. That is, the homing direction can be analyzed as a moving direction to decrease the score differences obtained from a set of Haar-like features.

If there is a snapshot image I available, the matching score $S(I,M)$ for a given Haar-like

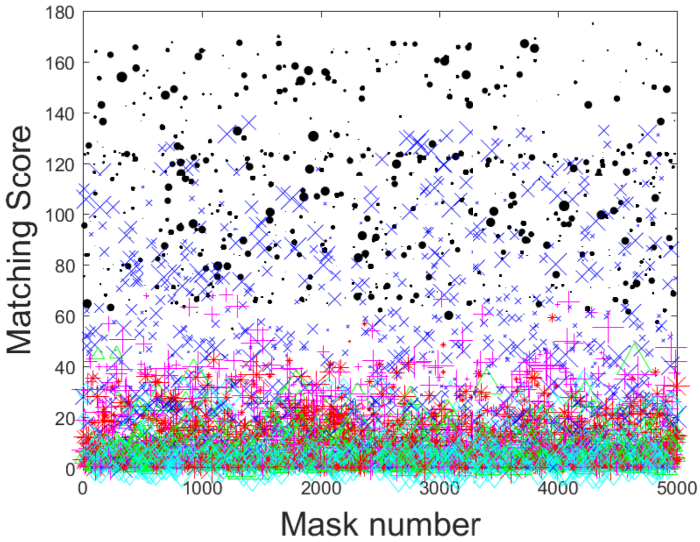


Figure 4.12: Matching scores for 5000 Haar-like masks. Six types of masks are tested and larger sizes of masks indicate higher matching scores; black dot for type 1, blue x for type 2, red star for type 3, magenta + for type 4, cyan diamond for type 5, green triangle for type 6.

mask M is calculated as

$$S(I,M) = |\sum_{i\in M+} I_i/m_1 - \sum_{i\in M-} I_i/m_2| \tag{4.8}$$

where I_i indicates the i-th pixel of the image and $m_1 = \sum_{i\in M+} 1$, $m_2 = \sum_{i\in M-} 1$ are the size of white and black regions, respectively. The score calculates the difference of the mean values of pixel intensities of white and black regions. We can easily see that the shape, location and size of the mask is important to get high score.

Many interesting landmarks are observed near the horizontal line. Random Haar-like masks are generated around the horizontal line. The bottom image in Figure 4.11 shows examples of well-matched cases for six types of masks. Type 1 can be matched well on a flat section or object. Type 2-3 are good for the boundary line matching, and Type 4-5 can be better for isolated segmented objects. Type 6 will be effective on the corner of objects.

Each mask has a configuration (θ, y) with size $(w,h) = (\Delta\theta, \Delta y)$ in the panoramic image that has 720 width (0.5 degree resolution) and 120 height pixels. For a snapshot image, 5000 Haar-like features were randomly generated. In this part, each mask has boundary in the size of generation. We generate them at random position around horizontal line within ± 10 pixels. Then we randomly pick its mask type and designate both width and height. This width is between 6 pixels and 120 pixels. Also this height is between 18 pixels and 60 pixels. Figure 4.12 shows a distribution of scores for the whole set of masks, and each mask has its own angular position in the snapshot image. We will see a large collection of masks contribute to see the visual characteristics of the snapshot image, more strictly speaking, a population of the mask score differences in the two snapshots characterize dominant change in the visual landmarks.

A set of Haar-like masks is applied to the two snapshot images, home (I_0) and the current view (I). Each type of mask has one or two rectangular regions, white ($M+$) or black ($M-$). Then a mask (M_j) can be represented by

$$\mathrm{M}_j(\theta_j, y_j, w_j, h_j) = \mathrm{M}_{j+} \cup \mathrm{M}_{j-}, \qquad \mathrm{M}_{j+} \cap \mathrm{M}_{j-} = \emptyset \tag{4.9}$$

where the j-th mask M_j has an angular position θ_j, y-axis position y_j, width w_j and height h_j. There is no overlapped area between white and black regions, and the center position of a mask is (θ_j, y_j).

We calculate the matching score $S(I, M_j)$ for each Haar-like mask M_j, using eq. (1), and a landmark vector $\overrightarrow{L_j}(I)$ is defined as a combination of the score and angular position of each mask. That is, its score is set as the vector length and the angular position as the vector direction. Thus, the landmark vector is written as

$$\overrightarrow{L_j}(I) = S(I, M_j)\hat{u}_j = (S(I, M_j)\cos\theta_j, S(I, M_j)\sin\theta_j) \tag{4.10}$$

where I is a snapshot image taken at a specific position, θ_j is the angular position for mask M_j and $\hat{u}_j = (\cos\theta_j, \sin\theta_j)$ is a unit vector for direction θ_j.

Based on the snapshot model, we can estimate the homing vector as a difference between the landmark vectors at the home location and at the current position.

$$\begin{aligned} \overrightarrow{H} &= \sum_{j=1}^{N} \overrightarrow{h_j} = \sum_{j=1}^{N} (w_j\hat{u}_j - v_j\hat{u}_j) & (4.11) \\ &= \sum_{j=1}^{N} (w_j - v_j)\hat{u}_j = \sum_{j=1}^{N} (\Delta L_j \cos\theta_j, \Delta L_j \sin\theta_j) & (4.12) \end{aligned}$$

where N is the number of masks, $w_j = S(I_0, M_j)$, $v_j = S(I, M_j)$ for the home snapshot image I_0 and the snapshot I at the current position, and $\Delta L_j = w_j - v_j$.

The above landmark model is similar to the weighted form of ALV model (Lambrinos et al., 2000) or the DELV model (Yu and Kim, 2012, 2011b,a). This model is an abstract form of landmark vector. The matching score is not proportional to the landmark distance, but large magnitude of ΔL_j can be observed along the moving direction from the current position to the home position, since much change of the visual image along the direction can happen. A set of masks around the angular position can greatly contribute to $\Delta L_j = S(I_0, M_j) - S(I, M_j)$, which will be effective to estimate the homing direction.

Figure 4.13 shows an overview of our algorithm. Random masks with a variety of types are generated and their matching scores are calculated over two snapshot images at the home location and at the current location. Each mask determines a landmark vector whose direction is equal to the angular position of the mask and the matching score of the mask over a snapshot is denoted as the length of the landmark vector. A collection of landmark vectors represent a snapshot image, and possibly the averaged landmark vector can be an representative vector for the snapshot at a given location. The difference of the two averaged vectors at the home location and at the current position can determine the homing direction, similar to the DELV model (Yu and Kim, 2012). Or each mask calculates the difference of the matching scores over the two snapshots and a collection of the differences for a set of masks can estimate the homing direction.

4.1.3 Navigation Using Relative Distance of Matching Scores

The above landmark model (Method 1) describes how to estimate the homing vector using the matching scores of Haar-like masks. The method is largely affected by the image contrast or brightness. We note the relative score differences of masks rely on how far away the two snapshots are. The score difference increases when the distance becomes larger. We separate the relative distance into two components, the x-axis or y-axis movement.

To handle the problem, Method 2 uses multiple snapshot images including the home image (I_0), an image near home along the x-axis (I_X) and another image near home along the y-axis (I_Y). Those images are compared with the image at the current position through Haar-like feature masks. The method first calculates the matching score difference of each mask M_j between a pair of snapshots. The sum of the matching score difference over a set of masks approximately estimates the distance between the two snapshot positions. The relative distances of matching scores among the snapshot positions can determine the homing direction. Similar with DID which shows best

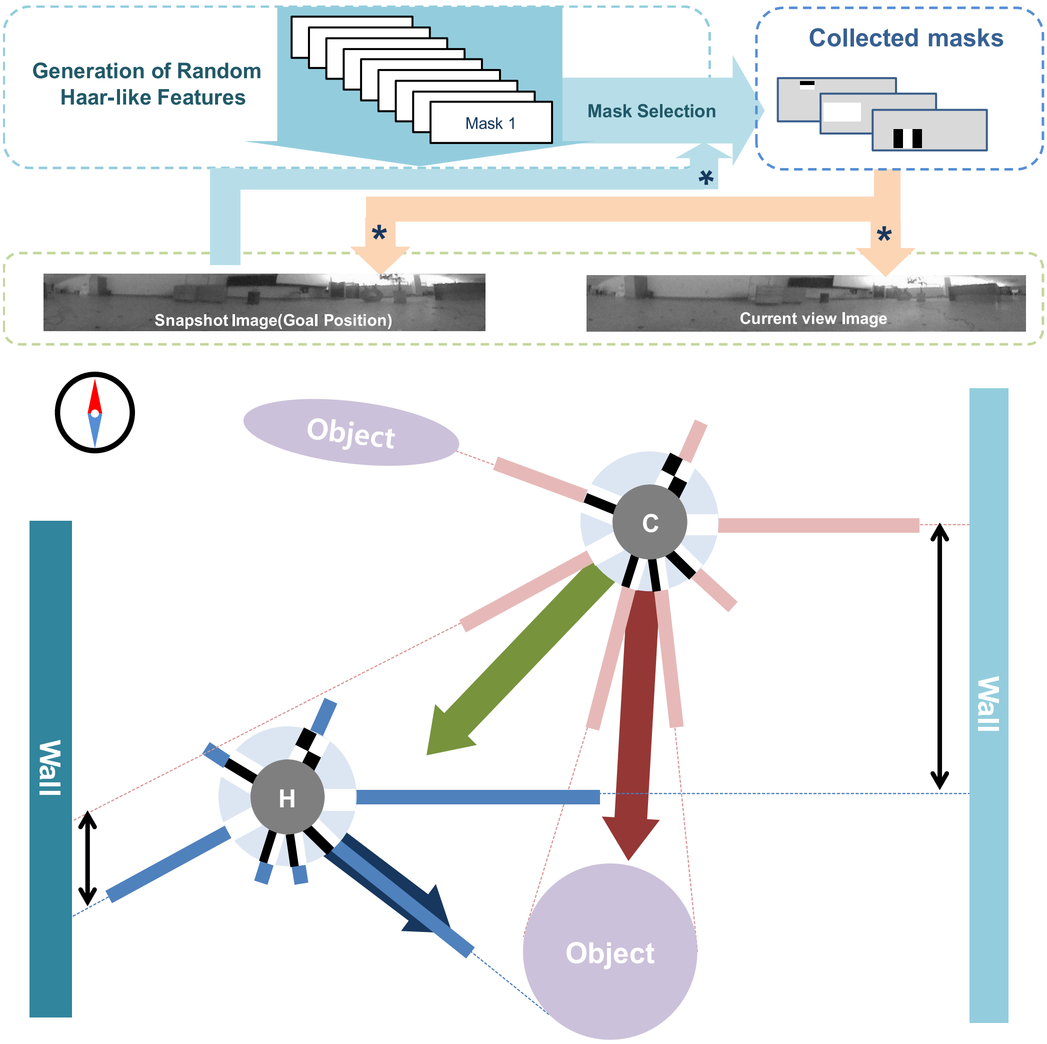


Figure 4.13: Overview of the Haar-like feature homing algorithm. Initially random masks are generated and the matching score of each mask over the home snapshot image can be calculated. Then a set of landmark vectors are derived from the mask scores. The same set of masks are applied to the current snapshot and another landmark vectors are obtained. The homing vector can be estimated from the two sets of landmark vectors. Lower figure shows a diagram of landmark vectors at the home location (H) and at the current location (C). The matching score for each mask becomes the length of the mask landmark vector. The average of landmark vectors are represented as blue arrow and red arrow, respectively. The difference of the two averaged vectors makes homing direction, displayed with green arrow.

performances in explaining insect's navigation, we try to mimic multiple images to robustly find homing direction. As mentioned in introduction, we think that learning sequences of insect's homing can be directly approximated gathering multiple reference images around home position. Therefore, based on image distance assumption, we use three reference images in method 2.

The homing direction can be estimated as follows:

$$X_{diff} = \sum_{j=1}^{N} |S(I_X, M_j) - S(I, M_j)| - \sum_{j=1}^{N} |S(I_0, M_j) - S(I, M_j)| \tag{4.13}$$

$$Y_{diff} = \sum_{j=1}^{N} |S(I_Y, M_j) - S(I, M_j)| - \sum_{j=1}^{N} |S(I_0, M_j) - S(I, M_j)| \tag{4.14}$$

$$\overrightarrow{H} = (X_{diff}, Y_{diff}) \frac{1}{\sqrt{X_{diff}^2 + Y_{diff}^2}} \tag{4.15}$$

where the homing vector ($\overrightarrow{H}$) is determined by the ratio between the x-direction difference and the y-direction difference.

The sum of score differences over a set of Haar-like masks for a pair of snapshot images can derive the relative distance between the positions for the snapshot images. Equation (6)-(7) is a simple but reasonable measure to determine the relative difference of the distance along the two different directions.

Figure 4.14 shows an overview of our second model. Random masks with a variety of types are generated and their matching scores are calculated over images including three reference images at the home location and the other at the current location. There are differences in the sum of mask scores based on image distances and the ratio between three images and current one are directly applied into finding homing direction. According to method 1, the combination of Haar-like masks can represent the visual information in those images and we try to find local gradient by using multiple reference images.

We propose two methods using Haar-like masks. One is to utilize the matching score of masks and the other is to estimate the relative distance of matching scores for the current position and home, through a set of masks. The second model needs two extra images near home.

In this part, we introduce process of our algorithm according to order of progress. We use only vision data using camera and the robot with this camera is located in the static experiment area. Later part is composed of visual mask modelling, image process, alignment, arrangement, scoring, landmark vector generation and finding homing direction. Also we add proof of global convergence in last part.

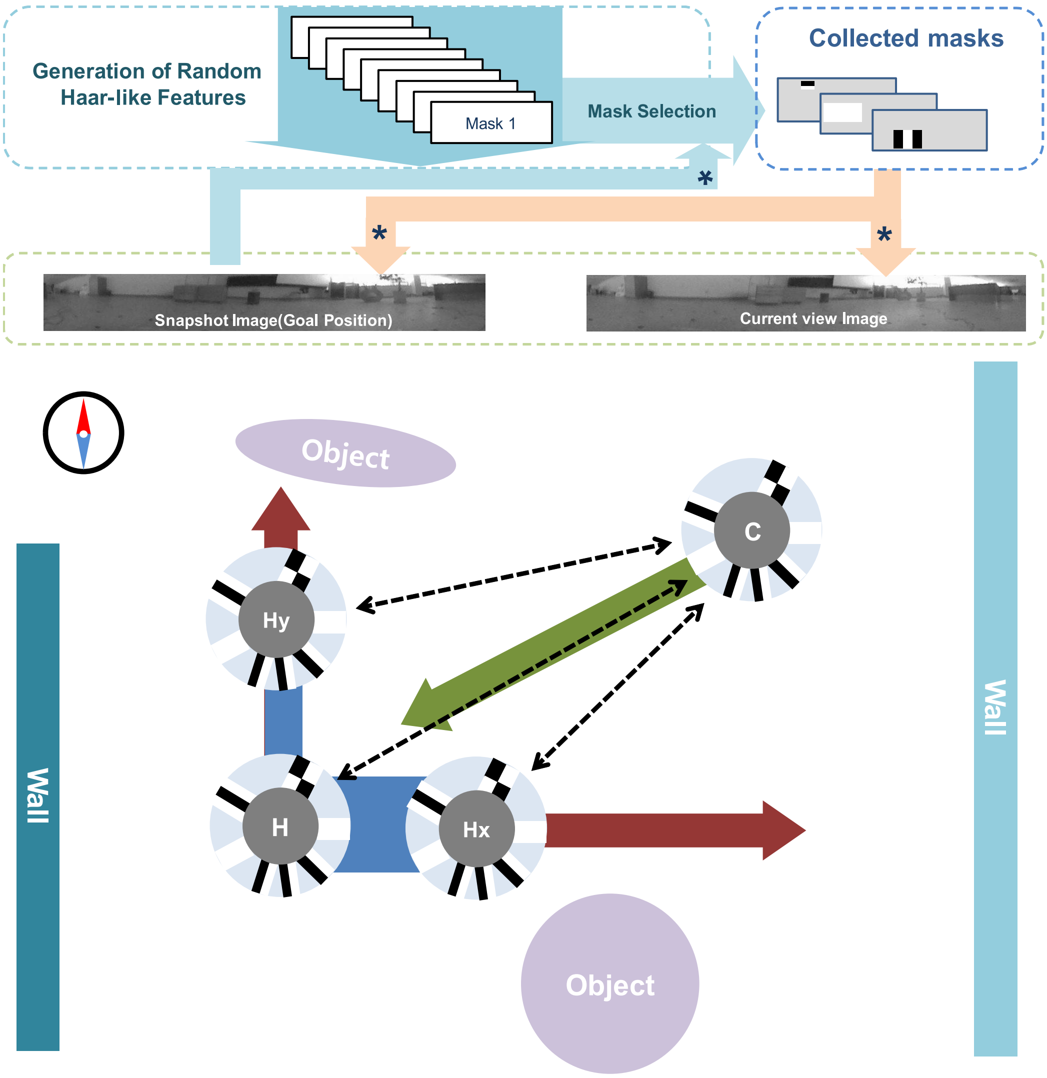


Figure 4.14: Overview of the Haar-like feature homing algorithm. Initially random masks are generated and the sum of matching scores of each mask over three reference images (H, H_x and H_y) can be calculated by applying the same set of masks. Each summation is directly proportional to geometric distance and the homing vector can be estimated from the differences of two terms. It shows that changes between pairs with the width of blue links and overall length as red arrows, respectively. The ratio between two pairs makes homing direction, displayed with green arrow.

4.2 Experiments and Results

4.2.1 Simulations

For the simulations, we considered various environments containing only cylindrical objects to comply with the abovementioned assumption. Figure 4.15 shows the corresponding results, where we considered different sizes, colors, locations, and number of landmarks. Overall, small errors can be perceived from the obtained homing directions, but global convergence is suitably satisfied.

In addition, some unsolvable cases are illustrated, but they are expected as determined by the convergence proof in Appendix. Nevertheless, these cases are uncommon in a real world scenario, and they are less severe than those that might arise from algorithms such as the ALV method (see details in Appendix).

4.2.2 Experiments Using Visual Masks

4.2.2.1 Visual navigation using only visual masks

The most basic part of the proposed method is the visual mask model. Hence, we evaluated its operation. Figure 4.16 shows the activity of visual masks corresponding to two images. The lines superimposed to each image represent the activity of the visual masks corresponding to each region along the horizontal line. We aimed to efficiently use the difference between the activity patterns to determine the homing direction. The left graph in Figure 4.17 illustrates the homing direction using only visual masks. It can be seen that the direction vectors mainly point to the home position, but additional processing is required to improve performance.

4.2.2.2 Visual navigation using mask set with matching

Next, we evaluated the mask operation integrated to the proposed matching algorithm. The homing directions are depicted in the right graph of Figure 4.17. The algorithm behavior is still inaccurate, but the inclusion of the matching process shows a considerable improved compared with the previous evaluation, shown in the left graph. However, the overall performance needs to be further improved.

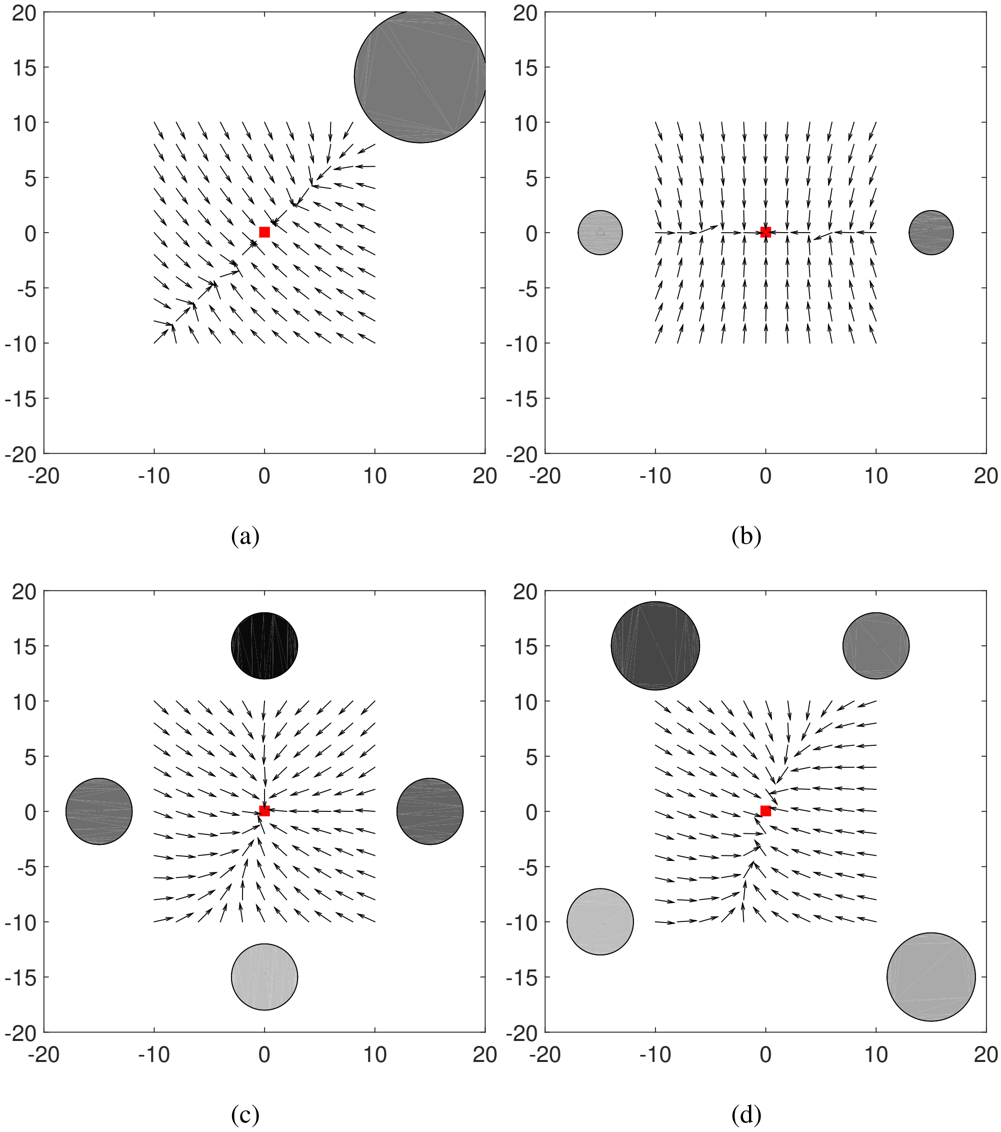


Figure 4.15: Simulation of homing navigation for different environments. The red square and circles represent the home position and landmarks, respectively. We also considered the landmark colors and a white background. The arrows indicate the homing direction at the corresponding positions.

4.2.2.3 Visual navigation using multilevel mask set with matching and filtering

Finally, we evaluated the complete proposed method, including the multilayer mask structure, connection weight matching, and filtering.

First, we evaluated the effect of the number of mask levels. Figure 4.18 shows the homing directions according to the number of mask levels. We tested the method using 1 (left graph), 6 (middle graph), and 9 (right graph) mask levels. The difference in the

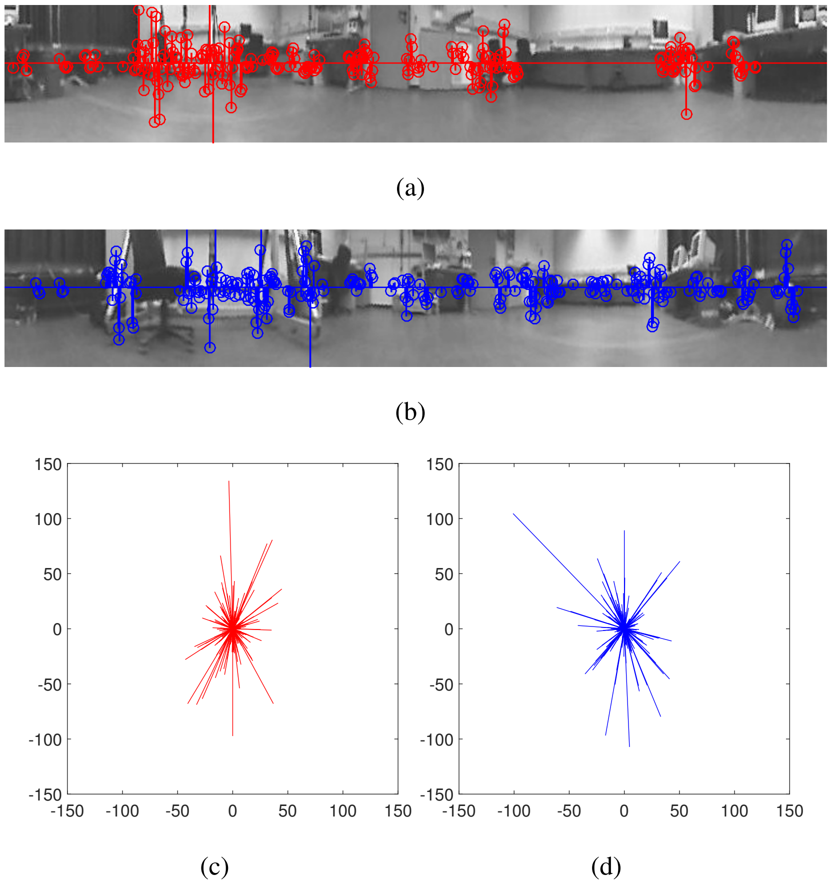


Figure 4.16: Landmark vector generation. (a) and (b) are taken from near positions. Only one type of visual masks along the horizontal line were considered. The lines over the pictures represent the firing rate of visual masks at the corresponding locations. The landmark vectors from the (a) and (b) are represented using (c)red and (d)blue lines, respectively.

homing directions is noticeable, but they are not significant. However, the calculation time notably increases according to the number of mask levels. Therefore, we should use a moderate number of levels (e.g., two or three).

Next, we evaluated the effect of mask size. For this experiment, we varied the basic mask size. Figure 4.19 shows the homing directions according to three different mask sizes. Either using a very small (left graph) or large (right graph) mask size generates inaccuracies in the calculated homing directions.

We also evaluated the effect of the amount of selected matching masks after filtering. Figure 4.20 clearly shows that the highest amount of selected masks (i.e., 40%) provides much better homing directions than the lower rates. However, higher rates do not

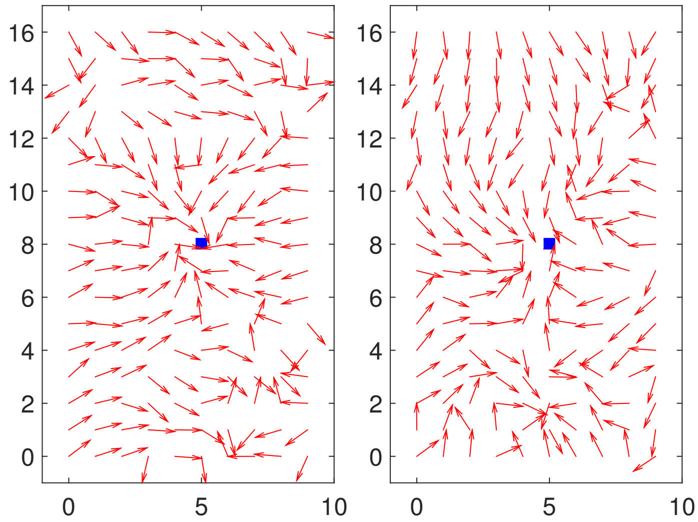
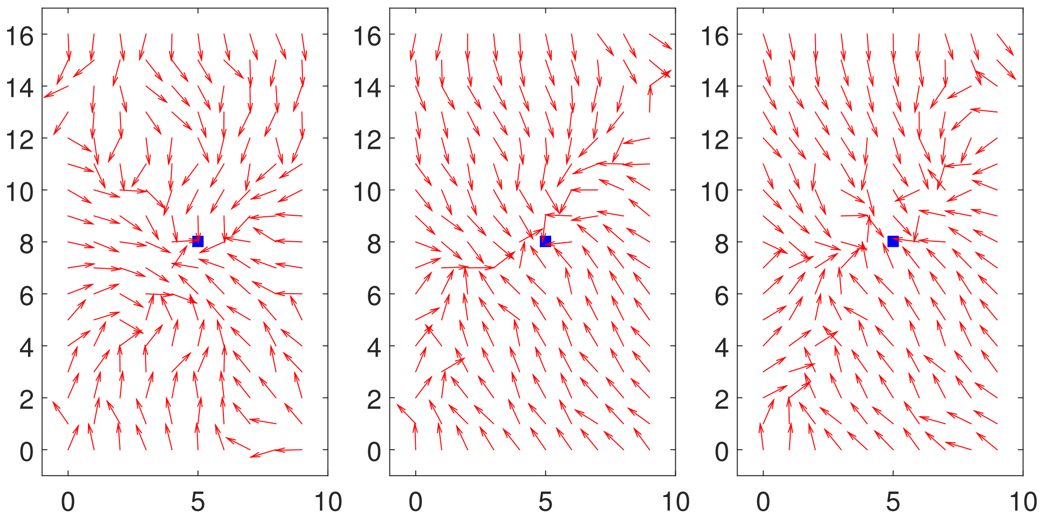
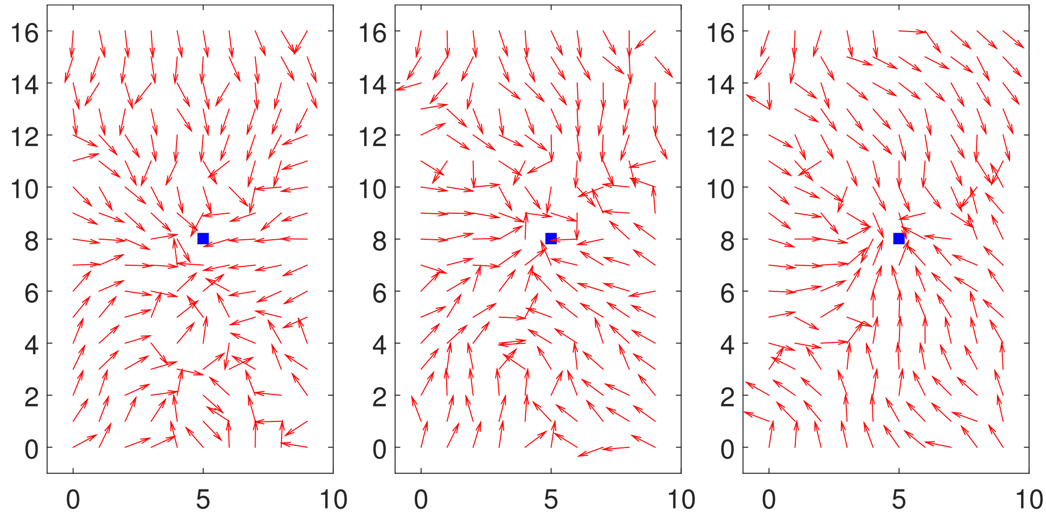


Figure 4.17: Homing directions from experiments using only the visual masks and combining these results with the matching algorithm. The blue square and red arrows represent the home position and the calculated homing directions at the corresponding positions, respectively. The left graph shows the results using only the visual masks, and the right graph adds the matching process.



(a) (b) (c)

Figure 4.18: Homing directions obtained from the proposed method considering the effect of the number of mask levels. The blue square and red arrows represent the home position and the calculated homing directions at the corresponding positions, respectively. We considered (a) 1, (b) 6, (c) and 9 mask levels.



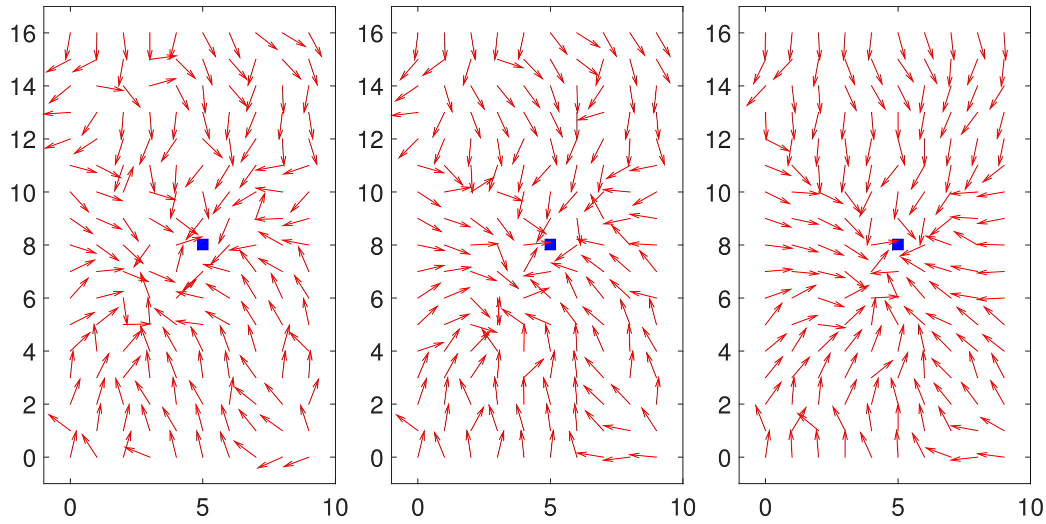
(a) (b) (c)

Figure 4.19: Homing direction obtained from the proposed method considering the effect of mask size. The blue square and red arrows represent the home position and the calculated homing directions at the corresponding positions, respectively. We considered masks located (a) every 1°, (b) 2° and (c) 4°.

provide better results than the 40% rate. The abovementioned experiments were useful to evaluate variations on different parameters of the proposed method.

Finally, we evaluated the proposed method according to the home position. Figure 4.21 shows the homing directions for three home positions. The results suggest that the homing navigation based on the calculated directions is accurate, provided that the home position is not very near to the objects.

In this section, we show general results that reflect the performance of the proposed method according to the parameter and home position variations described in the previous section. Figure 4.22 shows the effect of the number of levels in the top-left graph, and we can see that not many levels are required as the performance remains stable. Hence, increasing the number of levels does not notably improve in accuracy, but a single level is not for successful homing navigation. In fact, a single level produces severe errors in near site and the overall homing pattern is contaminated. Thus, we suggest using 3 levels for suitable navigation. The top-right graph shows the effect of mask size, where we can verify the minimum error cases, and we used 4° as the most suitable solution. The bottom-left graph shows the effect of the amount of selected masks after filtering. In this case, using rates above 40% does not increase performance, but



(a) (b) (c)

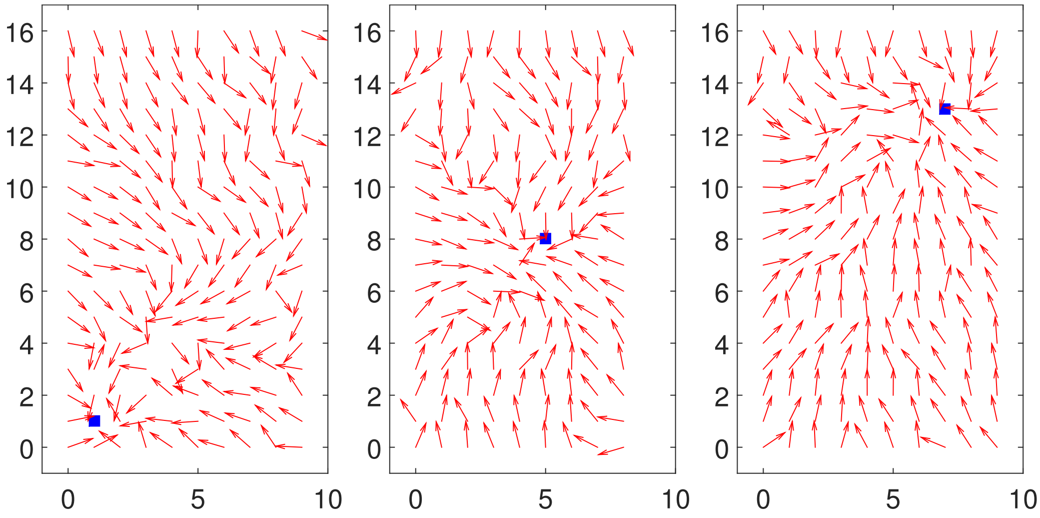
Figure 4.20: Homing direction obtained from the proposed method considering the effect of the amount of matching masks selected by the filter. The blue square and red arrows represent the home position and the calculated homing directions at the corresponding positions, respectively. We considered different amount of matching masks selected after filtering: (a) 6%, (b) 12%, and (c) 40%.

demands more computational time. From these results, we tuned the proposed method by using 2 levels, mask size of 4°, and 40% of the highest matching masks. Finally, the bottom-right graph verifies the angular error of the tuned version of the proposed method. We considered the point (5,5) as the home position and obtained a 100% of homing accuracy. Furthermore, the mean angular error is below 30°.

4.2.3 Experiments using Modified Mask Generation

4.2.3.1 Visual navigation using randomly generated masks

Next, we expand our prior model using visual masks to use various types of masks with random generation. In this part, we test five indoor environments, 'ourlab' from our lab environment and Vardy's dataset environments 'a1original', 'screen', 'arboreal' and 'hall1'. The first landmark model uses only one home snapshot while the second model uses three snapshots around the home location, including the home snapshot image. Each environment has panoramic snapshot images in the grid with regular intervals.



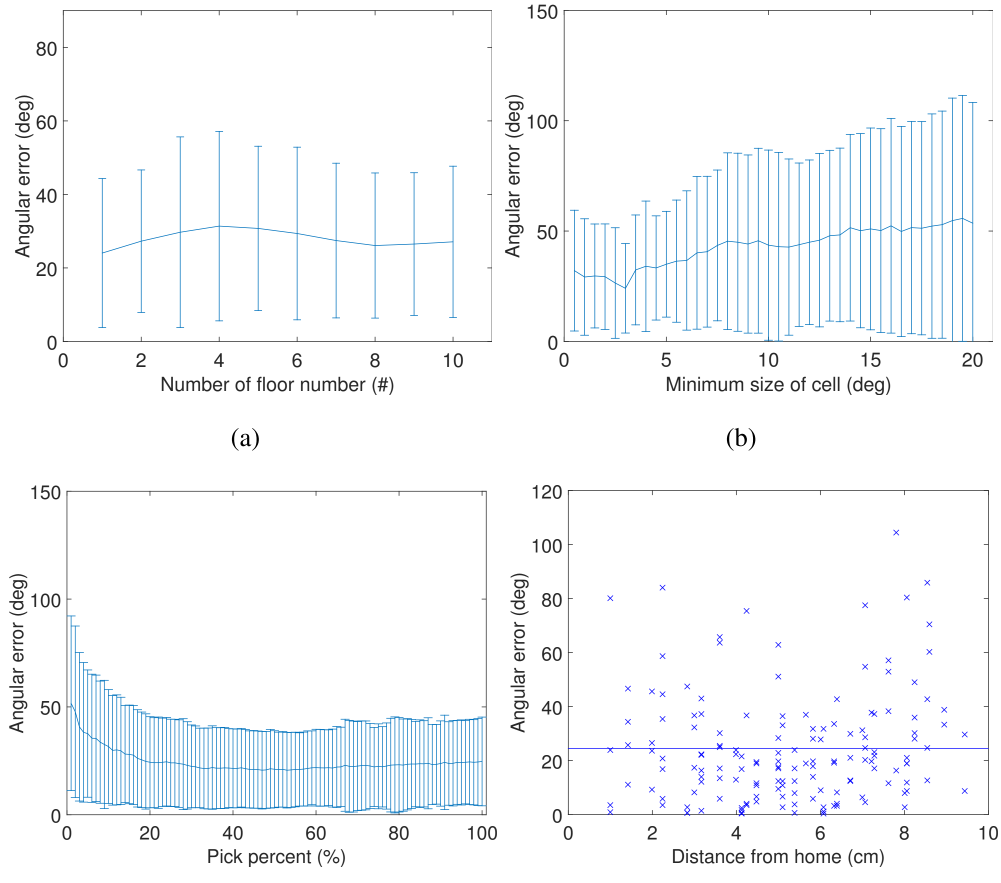
(a) (b) (c)

Figure 4.21: Homing directions according to different home positions. The blue square and red arrows represent the home position and the calculated homing directions at the corresponding positions, respectively. We considered three home positions: points (a) (1,1), (b) (5,8), and (a) (7,13).

We first test Method 1 based on Haar-like feature scores, where one home image is used. The whole set of Haar-like features has 5000 randomly generated masks with varying sizes, random locations and types.

An example of matching score differences for a collection of random masks is shown in Figure 4.23. The score differences within a window size of 90 degrees are accumulated into each bin and it roughly estimates the best homing direction (90°, the weighted average of histogram as a population coding). If the test position (5,7) is close to the home location (5,8), more symmetric distribution is observed. Even at a far position (5,1), the distribution of matching score difference can determine the homing direction. We also note that the sum of bin heights has information of geometric distance to home. This idea will be used in Method 2.

Figure 4.24 shows the homing performance, depending on the number of random Haar-like masks. Even 100 masks can roughly estimate the homing direction at most of positions, although there are large angular errors for homing direction at some positions, especially at the bottom left area. With 500 masks, the homing performance was much improved. More masks were assigned to reflect well the environmental situation in that area. A sufficient number of random masks can guide homing well and the angular er-



(a) (b)

(c) (d)

Figure 4.22: Performance graphs according to the different parameters(a:number of floor, b:minimum size of cell, c:pick percent) evaluated in the previous section. The (d) graph considers distance from home with respect to point (5,5). The blue lines represent the mean values.

rors can be reduced greatly. As the number of random masks increases, the homing performance can be improved up to a limit.

The angular errors for homing can be calculated as the difference between the desired homing direction and the estimated homing direction. The desired direction is directly drawn from the current position to the home location. Figure 4.25 shows the angular errors depending on varying number of masks. The set with 1000 masks or more shows stabilized homing performance as we observed the same result in Figure 4.24. The limit of homing performance may be related to the scope of the location, size or type of varying masks.

Next, we applied our landmark model to two different environments, ‘a1original’ and

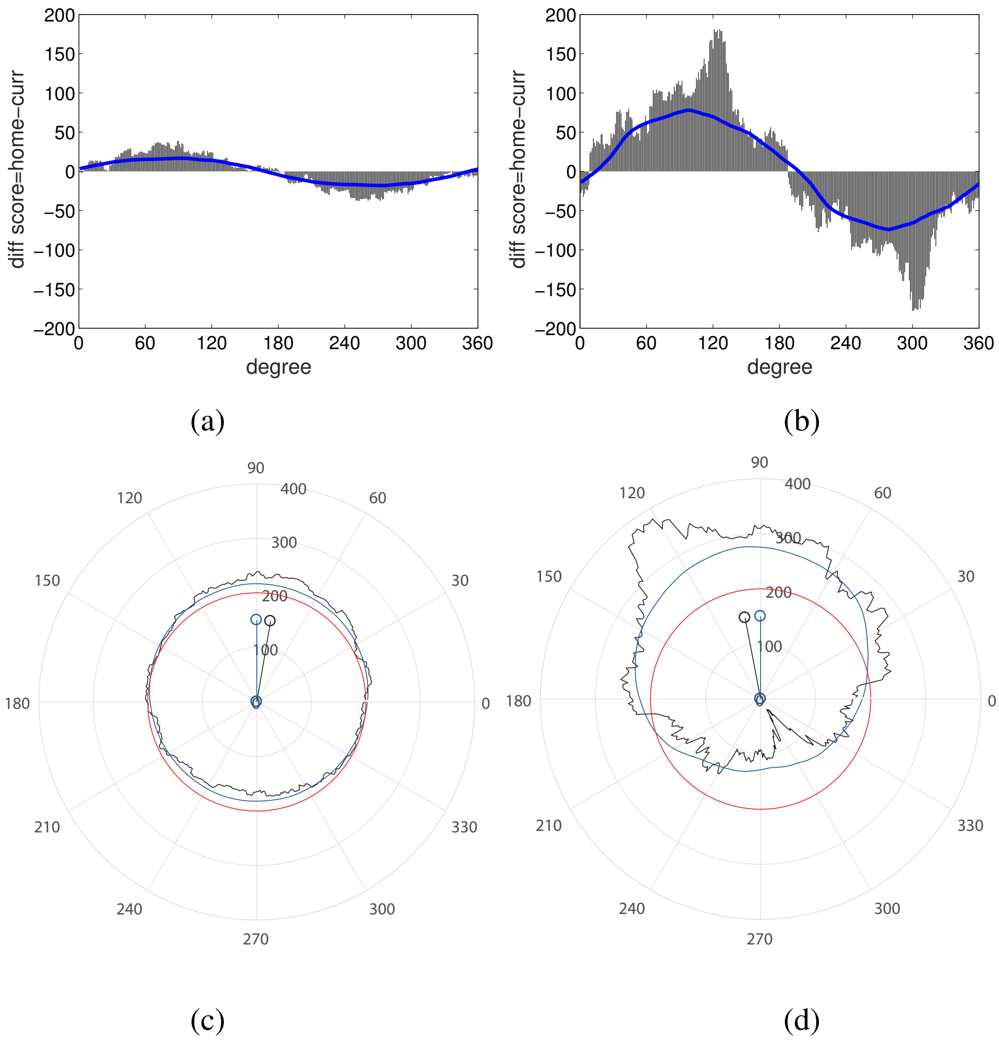


Figure 4.23: An example of histogram of matching score differences over a pair of snapshot images (blue curve indicates a low-pass filtered result over the histogram) (a) close distance for the two snapshots (b) far distance for the two snapshots Circular plot for each case. Based on center, it plots matching score in each direction and plot homing direction. Blue arrow is optimal direction and black is calculated one. (c) close distance for the two snapshots (d) far distance for the two snapshots

'screen'. In Figure 4.26, the model shows a reasonable pattern of homing directions in the environment 'a1original', but the flow of homing directions is not successful at the left side of the environment 'screen'. The 'screen' environment has some occlusion of visual landmarks, which makes the problem difficult. The suggested landmark model with Haar-like masks assumes that all the landmarks should be viewed in any position. Violating that condition may degrade severely the homing performance.

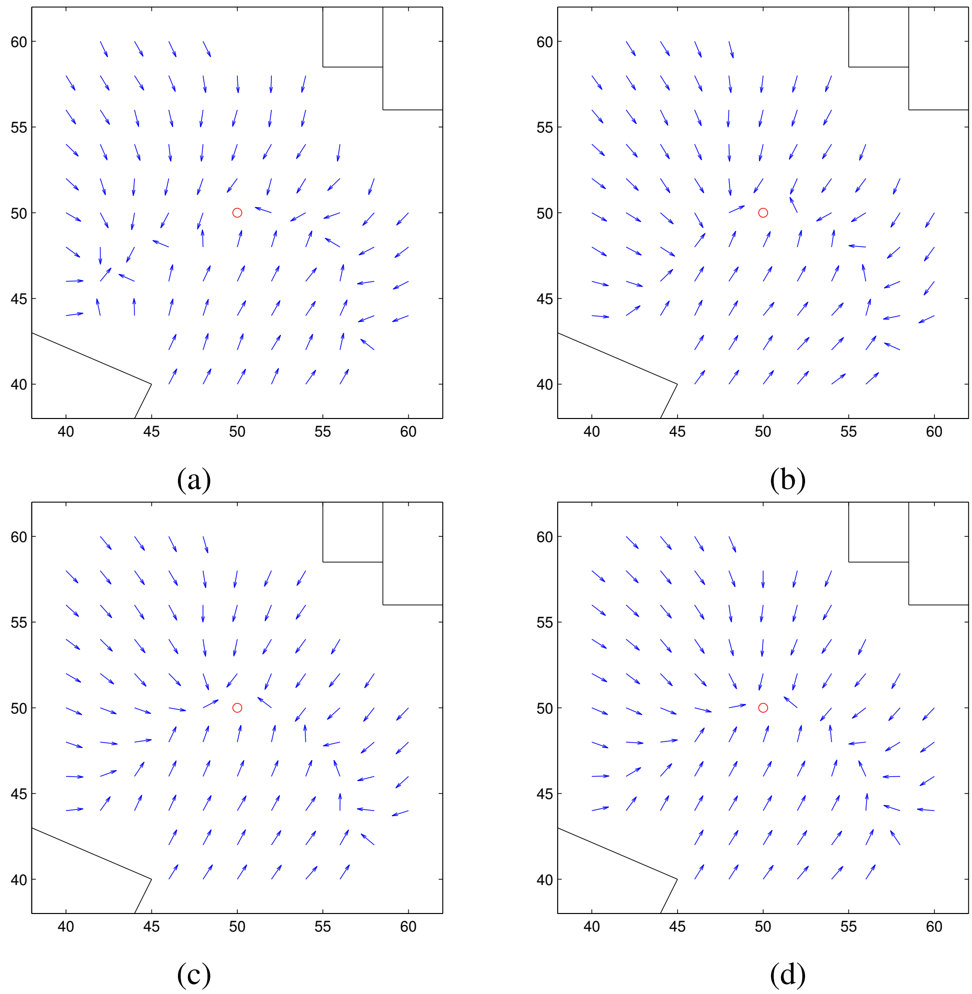


Figure 4.24: Homing performance in 'ourlab' environment using Method 1 with varying number of Haar-like masks (a) 100 masks (b) 500 masks (c) 1000 masks (d) 5000 masks; red circle indicates the home position and each arrow the homing direction.

4.2.3.2 Visual navigation using relative distance model

In a similar way to the above experiments with Method 1, we tested Method 2 based on the relative distance of matching scores. This method uses three snapshot images around the home location and thus three comparisons with the snapshot at the current position are required. The matching score difference between each pair of snapshots can determine the homing direction.

Figure 4.27 shows the sum of matching score differences for the two snapshots at the home location and at an arbitrary position and it is roughly proportional to the distance between the two positions. The image difference depends on the environment, but if a target position is fixed, the geometric distance can be estimated by the matching score

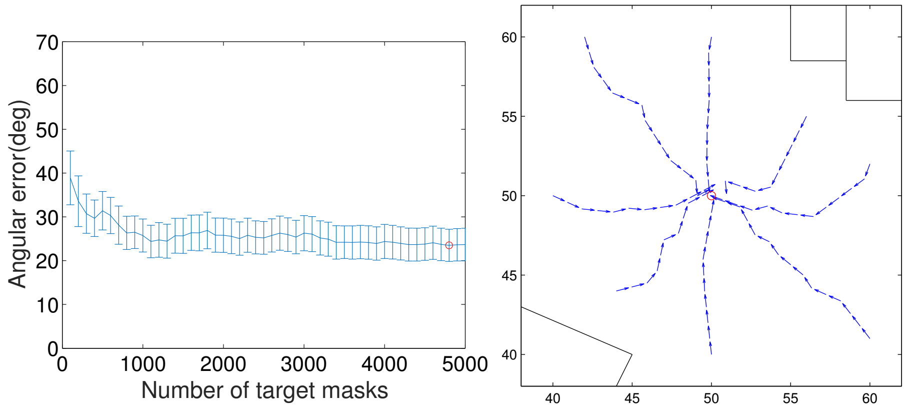


Figure 4.25: Left : Angular errors with Method 1 depending on varying number of Haar-like masks for environment 'ourlab' (error bars indicate 95% confidence intervals assuming t-distribution). Red circle means min value in 95 percent within error distribution Right : Homing route test from various location.

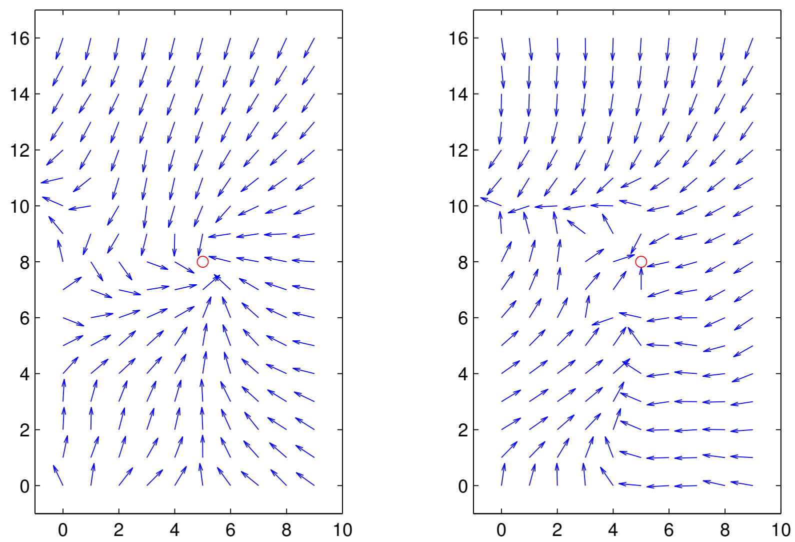


Figure 4.26: Homing performance in environments 'a1original' and 'screen' using Method 1 with 1000 random masks; red circle indicates the home position and each arrow the homing direction.

difference at an arbitrary position. As shown in Figure 4.27, the sum of score differences changes within a variance for a fixed distance and four spots in the grid have the same distance from the home location (four x marks are available for a fixed distance in the figure). For small distances, the variance of the measure is small, and at a far distance, the variance becomes large. It indirectly supports that neighboring positions, or positions close each other have more similar measure values. Deciding the homing direction at a given location is involved with comparing the snapshot at the position and

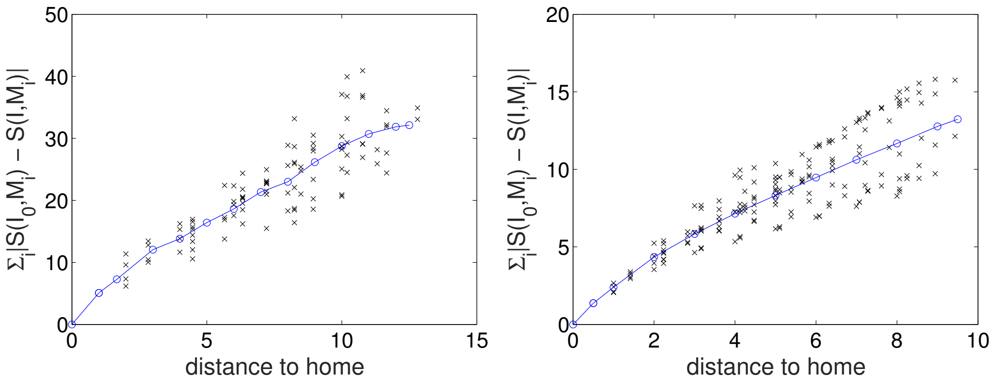


Figure 4.27: Sum of matching score differences over a pair of snapshots vs. distance between the snapshot spots (one snapshot is the home snapshot image); blue line is the curve fitted to data samples (a) environment 'ourlab' (b) environment 'a1original'

three similar snapshots around the home location. Those image differences through a set of Haar-like masks can more precisely estimate the homing direction.

Figure 4.28 shows the homing directions with Method 2 according to the number of masks. With 100 random masks, many scattered direction patterns are observed. The set with 500 masks or more show stabilized homing performance, while there are large angular errors at a few positions near the bottom left corner. Method 2 shows more direct direction patterns in homing, compared to Method 1.

When we tested varying number of random masks, the angular errors with Method 2 are significantly lower than those with Method 1 as shown in Figure 4.29. More random masks can improve the homing performance, and the performance seems to be stabilized with more than a limit number of masks. Attached to that, we add one more experiment in Figure 4.29. The right one shows the homing route in various starting position. There are eight starting positions and we can check those output having successful homing patterns.

We applied Method 2 to the other four environments. Figure 4.30 shows the results for homing performance, and Method 2 more robustly operate to estimate the homing directions in most of positions. The method calculates the matching score differences with three snapshots near the home location. The relative difference over the three snapshots is not much influenced even when there are occlusion points or significant change of landmarks at the current position. The surrounding background images will more contribute to the sum of matching score differences.

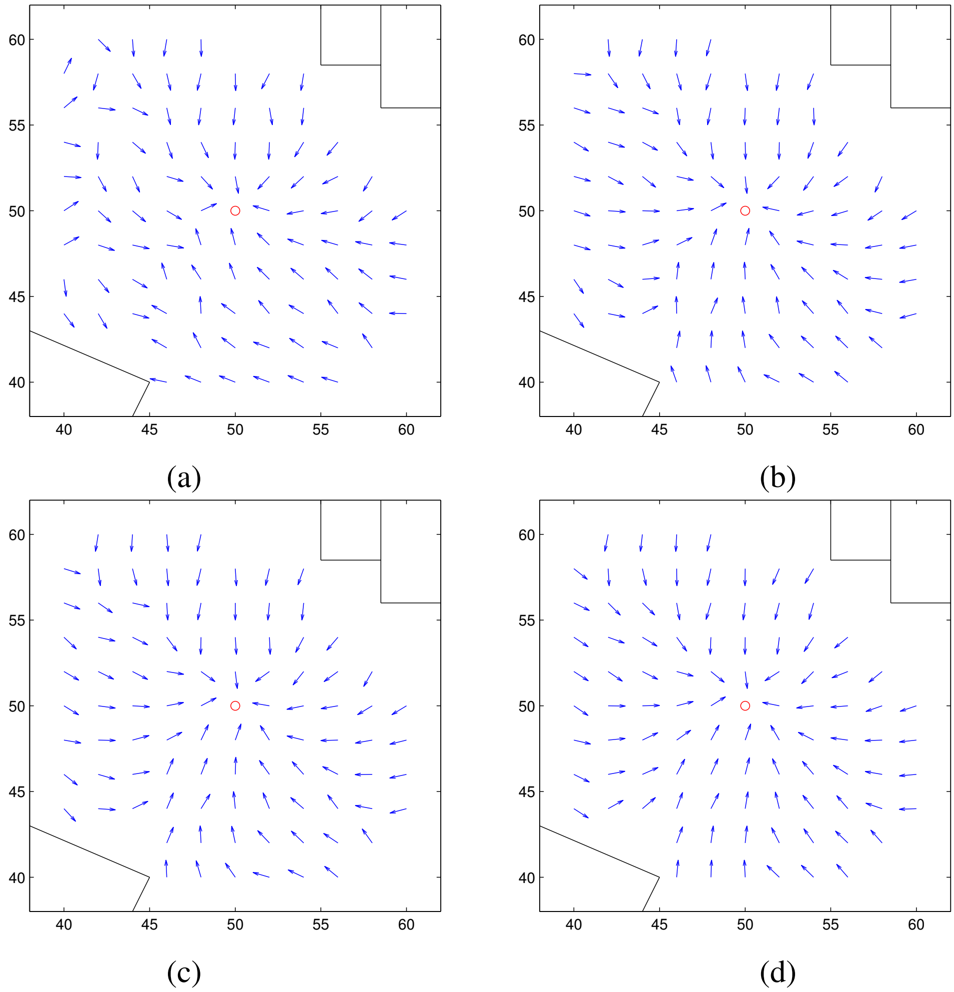


Figure 4.28: Homing performance in 'ourlab' environment using Method 2 with varying number of Haar-like masks (a) 100 masks (b) 500 masks (c) 1000 masks (d) 5000 masks; red circle indicates the home position and each arrow the homing direction.

Figure 4.31 shows the result for homing in the environment 'a1original' with varying number of masks. With a small set of masks, for instance, 100 masks, there are large angular errors at many positions. Too sparse masks in the angular space have difficulty in catching the landmark characteristics. As expected, more masks helps better homing performance. A large number of masks can read the environmental features better. It is notable that more direct paths to the goal position from an arbitrary position are observed using Method 2 than Method 1.

Figure 4.32 shows the homing errors depending on varying number of random masks in four different environments. We observe commonly that a small number of masks, for example, 100 masks are insufficient to read the environmental information. The set with 1000 random masks or more can sketch the visual cue needed for good hom-

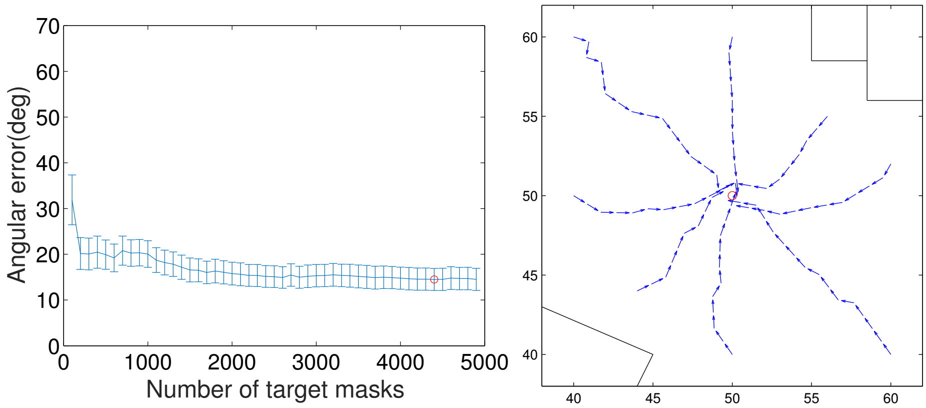


Figure 4.29: Left : Angular errors with Method 2 depending on varying number of Haar-like masks for environment ‘ourlab’ (error bars indicate 95% confidence intervals assuming t-distribution) Red circle means min value in 95 percent within error distribution. Right : Homing route test from various location.

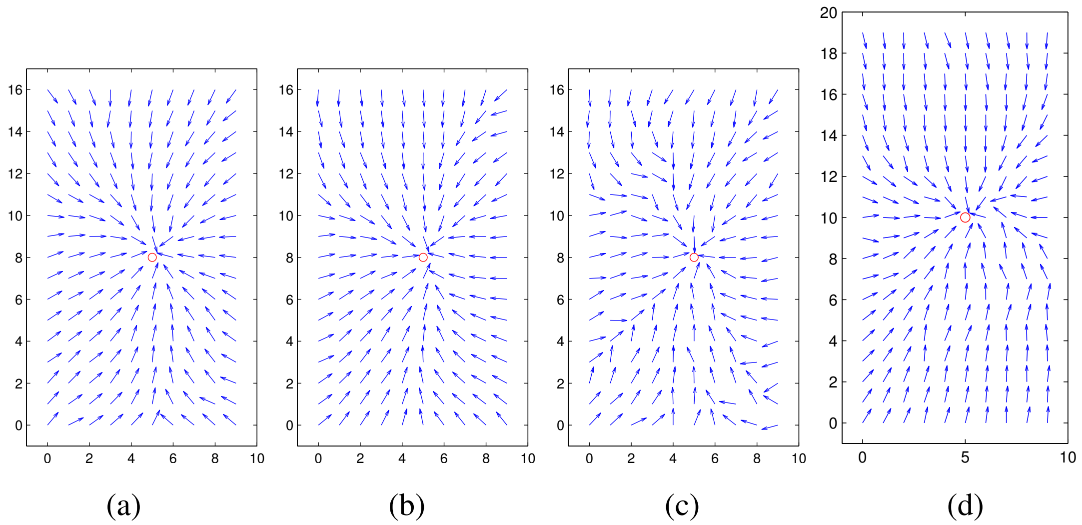


Figure 4.30: Homing performance with Method 2 in four different environments with 1000 random masks (a) ‘a1original’ (b) ‘screen’ (c) ‘arboreal’ (d) ‘hall1’; red circle indicates the home position and each arrow the homing direction.

ing performance. For the environment ‘arboreal’, much change of angular errors are observed when the number of masks increases. In contrast, the environment ‘hall1’ experiences very smooth change of angular errors when the number of masks increases, and a relatively small number of masks are sufficient to derive homing directions.

Table 1 shows homing performance for various environments. Method 1 shows good results in the environment ‘a1origin’, but it has worse performance in the environment ‘hall1’. Method 2 produces small homing errors for all the test environments.

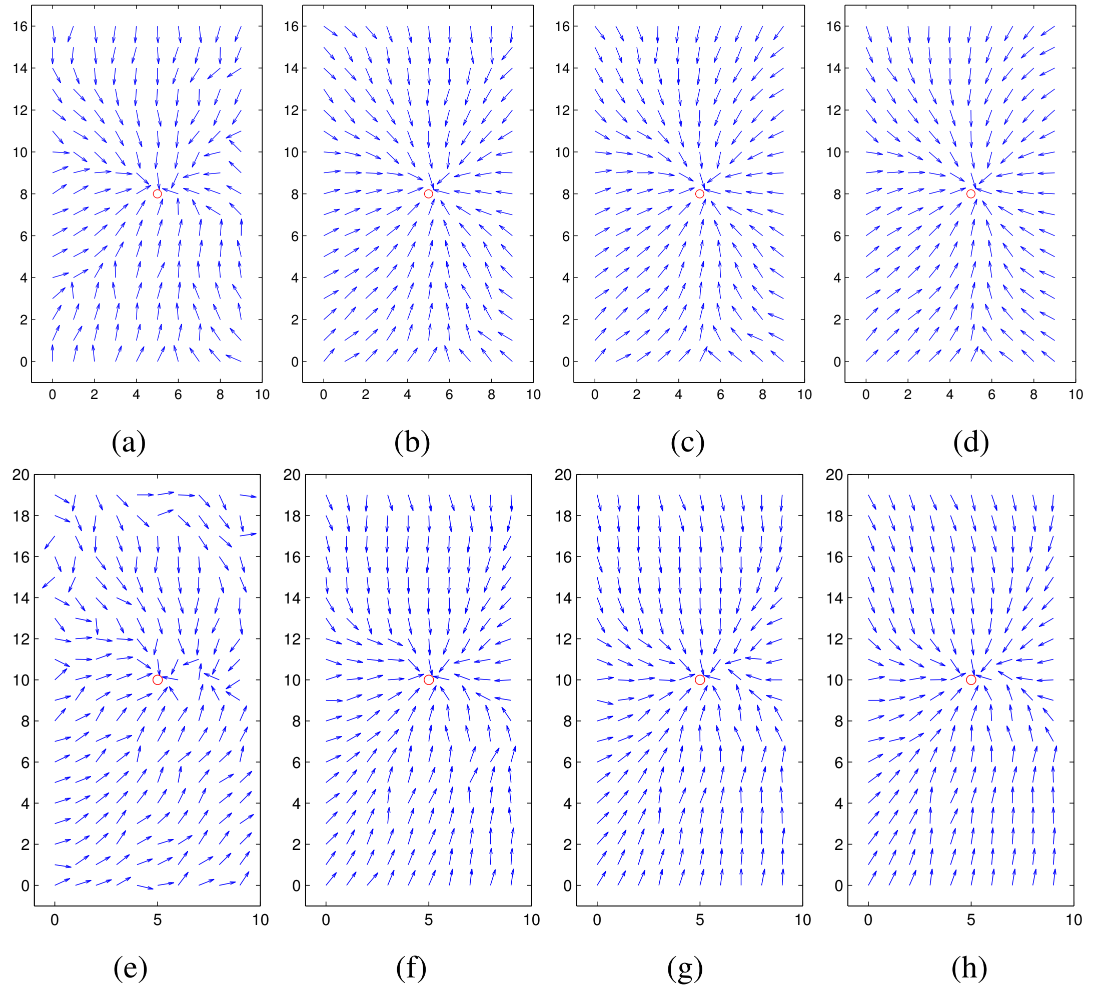


Figure 4.31: Homing performance in 'a1original' (a d) and 'hall' (e h) environment using Method 2 with varying number of Haar-like masks (a,e) 100 masks (b,f) 500 masks (c,g) 1000 masks (d,h) 5000 masks; red circle indicates the home position and each arrow the homing direction.

4.2.3.3 Comparison with other algorithms

Our snapshot models are compared with one of state-of-art techniques for visual homing navigation, the DID (Descent in Image Distance) method (Zeil et al., 2003; Möller and Vardy, 2006), MinWarping algorithm (Horst and Möller, 2017; Fleer and Möller, 2017), COMALV (Mangan and Webb, 2009; Hafner, 2001) and MCOMALV.

Figure 5.42 shows vector maps to compare our approaches with other four benchmark algorithms. Interestingly, Method 2 has more direct direction towards home than the other approaches. Table 1 shows homing performance for Method 1, Method 2, DID, MinWarping, COMALV and modified COMALV method for the environments. Except

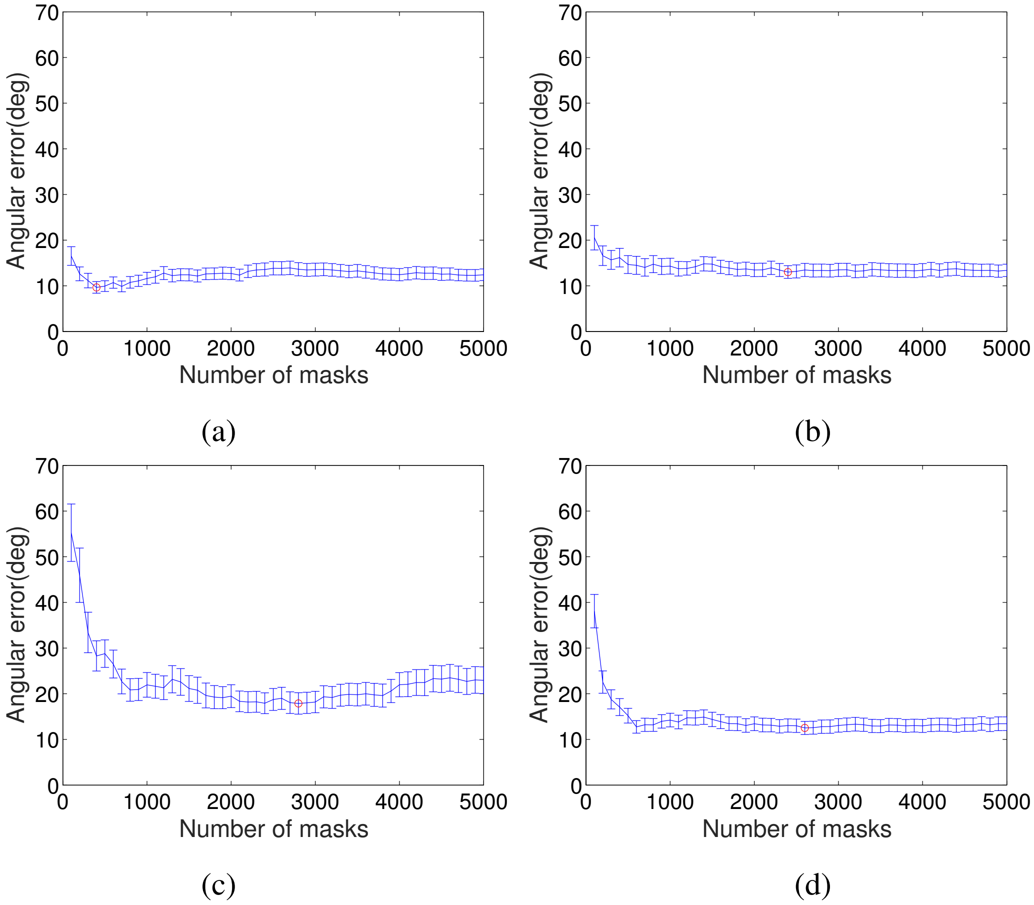


Figure 4.32: Angular errors with Method 2 depending on varying number of Haar-like masks for four environments (a) ‘a1original’ (b) ‘screen’ (c) ‘arboreal’ (d) ‘hall1’ (error bars indicate 95% confidence intervals assuming t-distribution) Red circle means min value in 95 percent within error distribution

for ’OurLab’ environment, MinWarping shows the best performances and our method 2 is next. Also, compared with the DID method using the whole pixels for snapshot comparison, but Method 2 uses Haar-like pixels for the image distance. Including all the environments with calculation time from the table 2, our method 2 shows more robust performance than others. Also, our model shows better output even when the test position is far away from the home location.

Figure 4.34 shows four experiments. The first one shows the length of homing vector to distance from home position. We find that the sum of differences of matching scores are proportional to geometric distances. Then the output shows reverse proportion because the distances between reference images are fixed. The second one shows the effect of blurring. Like insect perceives, we decrease the input image resolution

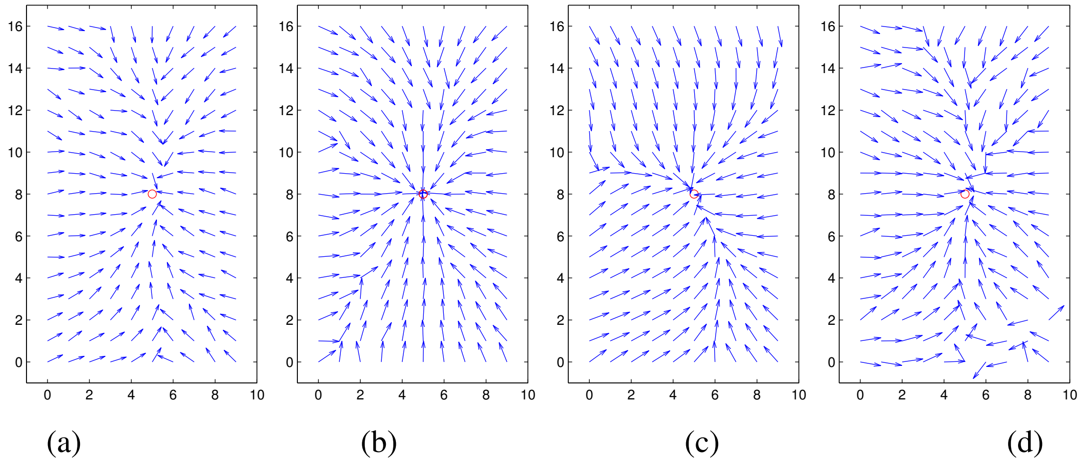


Figure 4.33: Comparison of DID, MinWarping, COMALV and MCOMALV method in vector map in ORIG environment (a) the DID method (b) the MinWarping method (c) the COMALV method (d) the modified COMALV method

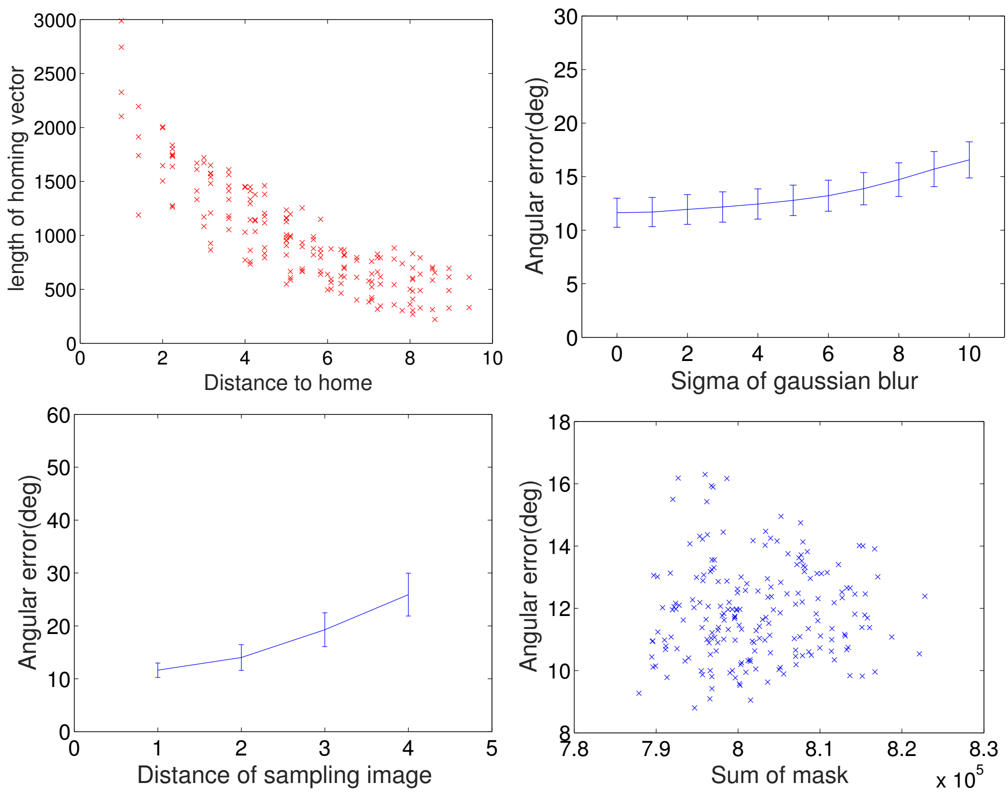
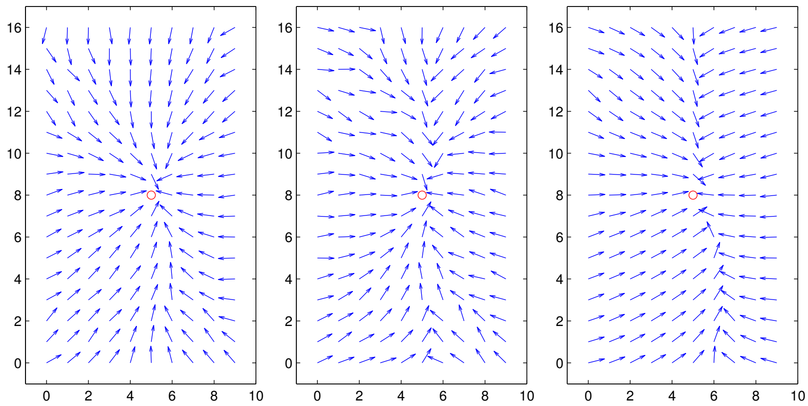


Figure 4.34: Four tests of method 2 with 1000 masks for length of homing vector to distance (a), the effect of down-sampling in homing accuracy with sigma in Gaussian blur (b), the effect of grid size in reference image selection (c) and the general score of generated mask sets in repetition test (d) according to the sum of each mask.

Table 4.1: Angular errors with various methods including ours in five environments (ε_θ is the absolute angular error and n is the number of test points; 1000 masks used,μ is mean of angular error and σ is the length of confidence interval in 95 percent. *p¡0.05).

		ourlab	a1original	screen	arboreal	hall1
	N	95	169	169	169	199
Method 1	error μ ($\pm\sigma$)	25 (±4.4)	16 (±1.8)	22 (±2.9)	27 (±3.0)	59 (±6.2)
	$0 \leq \varepsilon_\theta < 45°$	77%	96%	89%	78%	42%
	$45 \leq \varepsilon_\theta < 90°$	22%	4%	9%	21%	30%
	$90 \leq \varepsilon_\theta < 180°$	1%	0%	2%	1%	28%
	catchment area (%)	100%	100%	77%	68%	39%
Method 2	error μ ($\pm\sigma$)	20 (±2.6)*	9 (±1.2)*	13 (±1.4)*	18 (±2.4)*	13 (±1.4)*
	$0 \leq \varepsilon_\theta < 45°$	88%	99%	100%	96%	99%
	$45 \leq \varepsilon_\theta < 90°$	12%	1%	0%	4%	1%
	$90 \leq \varepsilon_\theta < 180°$	0%	0%	0%	0%	0%
	catchment area (%)	100%	100%	100%	100%	100%
DID	error μ ($\pm\sigma$)	25 (±3.0)	23 (±2.6)	19 (±2.0)	37 (±5.4)	21 (±1.7)
	$0 \leq \varepsilon_\theta < 45°$	94%	92%	96%	75%	95%
	$45 \leq \varepsilon_\theta < 90°$	6%	7%	4%	14%	5%
	$90 \leq \varepsilon_\theta < 180°$	0%	1%	0%	11%	0%
	catchment area (%)	100%	100%	100%	88%	100%
MinWarping	error μ ($\pm\sigma$)	61 (±9.5)	11 (±1.5)*	20 (±2.9)	16 (±2.4)*	31 (±4.2)
	$0 \leq \varepsilon_\theta < 45°$	45%	99%	91%	92%	81%
	$45 \leq \varepsilon_\theta < 90°$	25%	1%	8%	8%	13%
	$90 \leq \varepsilon_\theta < 180°$	26%	0%	1%	0%	6%
	catchment area (%)	67%	100%	100%	100%	95%
COMALV	error μ ($\pm\sigma$)	49 (±7.4)	19 (±2.0)	34 (±4.4)	44 (±6.3)	49 (±6.6)
	$0 \leq \varepsilon_\theta < 45°$	56%	98%	71%	64%	58%
	$45 \leq \varepsilon_\theta < 90°$	29%	2%	25%	21%	21%
	$90 \leq \varepsilon_\theta < 180°$	15%	0%	4%	15%	21%
	catchment area (%)	12%	100%	64%	51%	59%
MCOMALV	error μ ($\pm\sigma$)	35 (±5.4)	25 (±3.3)	19 (±2.9)	26 (±4.1)	24 (±2.2)
	$0 \leq \varepsilon_\theta < 45°$	72%	86%	93%	83%	90%
	$45 \leq \varepsilon_\theta < 90°$	25%	13%	6%	12%	9%
	$90 \leq \varepsilon_\theta < 180°$	3%	1%	1%	5%	1%
	catchment area (%)	96%	98%	99%	96%	99%

to find effect in homing accuracy. As haar-like mask has similar effect of average filter but large sigma can makes poor performances. The third one shows the effect in sampling distance in collecting reference images. In this part, we change the distance between home and other reference image location into grid size. Increasing distance in x direction is directly applied into homing performances. Then we can think that we make small distances between home and other references but there must be difficulties in catchment area that we can not find in this experiment. The fourth one shows the repetition of our model. According to the sum of mask region, our model has differences in accuracy of homing performances from 9 to 20. Also, we check that the



(a) (b) (c)

Figure 4.35: Comparison between our second model with best case, DID method and angle-fixed version. (a) vector map is the case using our second model, (b) with DID method and (c) with angle-controlled version of our model.

Table 4.2: Comparison of time consumption with six different methods in the environment 'a1original'

	Method 1	Method 2	DID	MinWarping	COMALV	MCOMALV
Calculating Time(sec)	0.3345	0.5721	0.0003	10.0158	0.00003	0.00005

time consumption of each methods through Table 4.2. In this part, we can check that our method 2 shows small time and not bad performance than other two benchmark algorithms.

Figure 4.35 shows three outputs. The left one shows the output using our second model and middle one shows the output using DID method. The right one is modified version of our second model to have equally distributed masks. In this part, we reorganize mask generation for equally distributed in horizontal line and select particular mask having highest score in equal direction. Then we can find a new output shows more similar to DID than our model.

4.3 Summary of Chapter 4

In this chapter, based on snapshot model of holistic method, we suggest and test a concept of visual masks as visual receptors to homing navigation. Our first method imitates the structure of bipolar cell by using the two types of haar-like features and applied generated multilayer masks on horizontal line to find homing direction. We find the convergence to home position by using both mathematical proof and experiments. Next, we made new models by adding randomness in mask generation process. With randomly generated masks, we can make homing pattern using the difference of matching score for each mask by estimating the distribution of landmarks in the environment. However, our models above cannot operate well in some environments having occlusion. Without feature extraction, our model cannot afford to filter it out by simpleness. Therefore, we apply using additional reference images like learning phase of insect, we try to find another robust homing method based on the image differences. Then we can demonstrate the effectiveness of the method in several environments.

Chapter 5

Visual Navigation Using Pixel-wise Information

In this chapter, we apply a concept of pixel-wise landmark that treat each pixel as landmark itself without object extraction. We try to find the basis principle of insect's homing navigation. For example, in vision system of insect, complex eye has poor resolution and it is not sufficient to make distance information. If it does not use object extraction by small number of neurons, it must use some other mechanism using receptor based homing navigation. We think a pixel as basic unit itself in homing navigation (Lee and Kim, 2016a). We call this pixel-wise landmark that operates like object and propose the new visual-based landmark navigation algorithm based on changes of these pixel-wise landmark information. Similar with ALV, It changes the snapshots of both home and current into pixel-wise landmark vector sets without object extraction. We show the effects of both distance estimated model and color model without distance. Through the experiments with mathematical proof, we investigate the convergence of our model. Then we made another one like ACV using landmark matching that is modified version of ALV. In this part, we use DTW algorithm that can match the time-series warped data to original one. Because time series concept is replaced with angular location then matched output of DTW is equal to the one of landmark vector matching. Therefore we add one idea about two static points which is not affected by movements and located at the opposite direction. Combining DTW with finding static points can make robust matching in pixel-wise landmarks. Finally, we use these matching to build landmark vector like ACV model and make robust visual homing algorithm. These researches are under the preparation for submission (Lee and Kim,

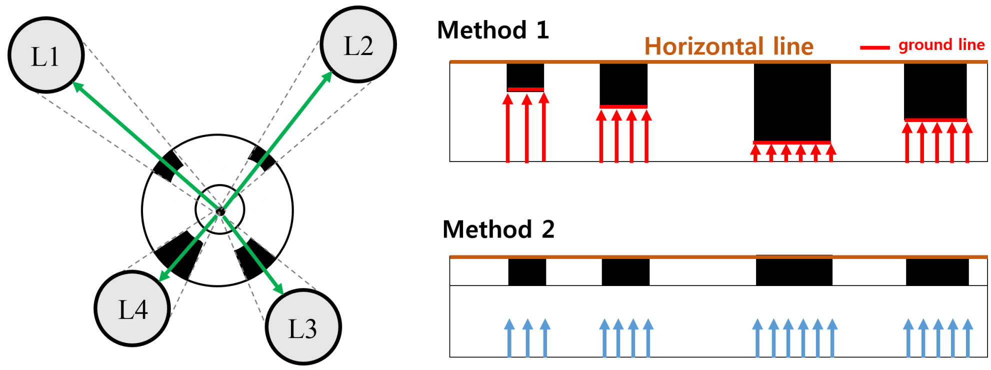


Figure 5.1: Type of landmark vectors. The first one (top-view) shows the arrangement of landmarks that are denoted as circles and the reflected omni-directional image represented by a circle with a black dot at the center in the first figure. The green arrows indicate the original landmark vectors. Without clustering, the pixel-wise landmark may be the one of either the red (Method using the estimated distance value based on the ground line) or the blue arrow (Method using only the angular position of the landmark pixel).

2017f,c). Detail procedures of our model is depicted in next part.

5.1 Pixel-wise Landmark Modelling

Prior to introducing our navigation algorithm, we explain our landmark modeling method. In this case, the landmark is an entirely observed object (in the experiment area) that can be considered as a geometrical feature in navigation. There exist numerous types of objects that can be used as one of the landmarks in our navigation. Our simple model did not include information of all of these things. Instead, we applied the idea of pixel-wise landmark; that is, we treated a pixel as the landmark itself without any clustering algorithm. In addition, we considered all objects as cylinders. This is because cylindrical shapes are not affected by the pose of the robot in the horizontal direction. If this modeling is appropriate, then other shapes can also be approximated.

Figure 4.9 shows the principle of the pixel-wise landmark method. The overall environment is flat, and both the floor and background are white. In addition, a brown line called the horizontal is an imaginary horizontal line that acts as a boundary between the background and floor. There are four landmarks (L1 L4) that are cylindrical and black in color. The input visual image is similar to the particolored circle of the first

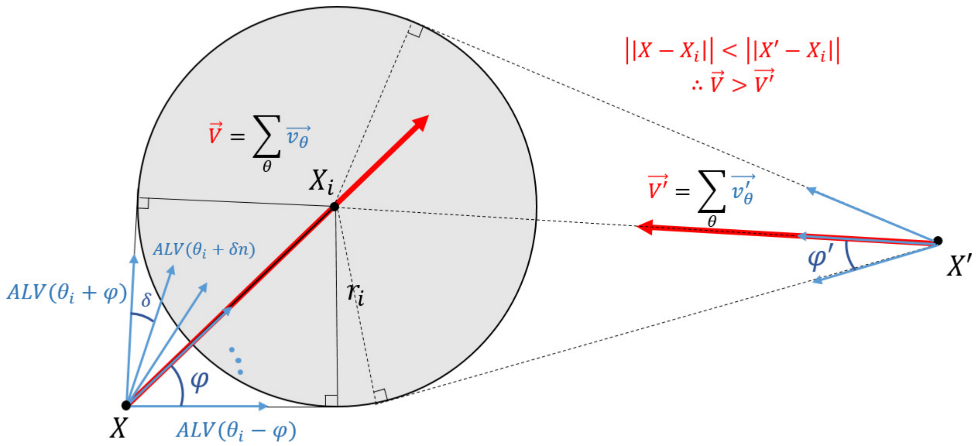


Figure 5.2: Principle and concept of the pixel-wise landmark method. The overall figure presents a top view. The gray circle indicates the cylindrical landmark, and X_i, the center with a radius r_i. Two observations are made at X and X'. The equal distance vectors called ALV point to the parts of object that are marked, as shown by the blue arrows. The sum of the blue arrows is represented by the red arrow that is not geometrically equal to the real landmark vector.

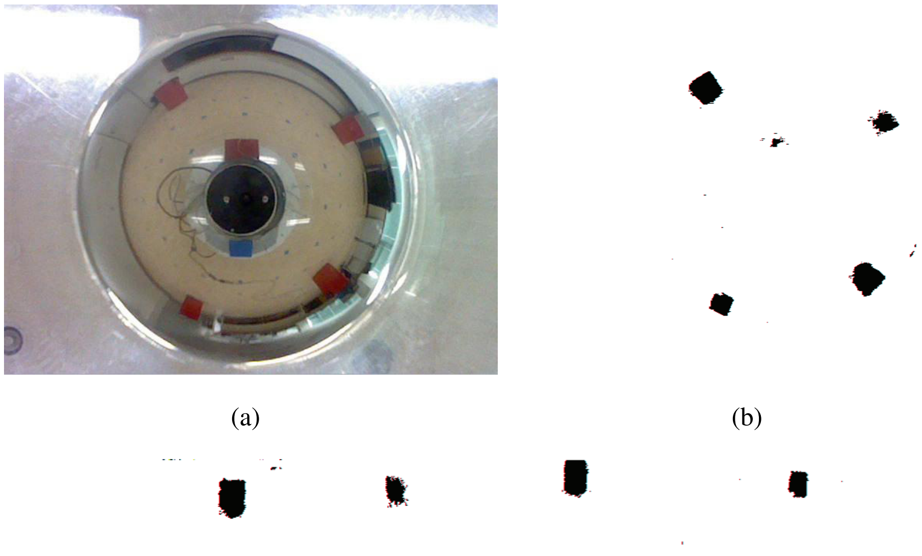
figure—this is similar to the vision of honeybees. Then, the green arrows are the general landmark vectors containing both angular location and distance information; these are obtained by using both the range sensor and clustering algorithm. Our robot does not have both the sensor and clustering algorithm, but we use pixel-wise landmarks.

Then, first method uses pixel-wise landmarks of different lengths based on the ground line marked as the red line in the black region. Second method uses only the angular position of the pixel of the black region. However, these pixel-wise landmark vectors (PLV) differ from the real ones. Therefore, we applied the assumption that all the objects are cylindrical in shape for the sake of approximation.

Similar to the ALV method Lambrinos et al. (2000), we use a summation of PLVs ($\vec{O}$) as representative of one location (X) and it can be calculated like

$$\vec{O} = \sum_{\theta} \vec{V_\theta} \simeq \lim_{\delta \to 0} \sum_{i} \frac{R_i}{r_i} \vec{u}(\theta_i)$$

where $\vec{u}(\theta_i)$ is unit vector pointing angle (θ_i) and it has weight (w_i) for each i-th object. r_i is distance from position (X) to landmark location (X_i) and size (radius) of each object is R_i. As shown in Figure 5.2, each weight of the object pointed by the vectors is not equal to the real one, and the value is inversely proportional to the real



(a) (b)

(c)

Figure 5.3: Example of simple landmark extraction a)The original omnidirectional image from the webcam, b)binary image, and c)binary-panoramic image

value. Therefore, an appropriate model should be built to use it. (Detail procedure is written in appendix)

5.1.1 Navigation Using Pixel-wise Landmark with Distance Estimation

In this paper, we propose the pixel-wise navigation based on various assumptions. This part shows the one of our model that uses estimated distance values. Prior to start, this model uses pre-designated landmark information that are for landmark extraction. Then, based on filtered image output, it tries to estimate distances to each angular direction based on ground line and outputs are applied into homing navigation.

5.1.1.1 Collecting landmark pixel

First, according to the assumption of fixed color characteristics of landmark objects, we can check whether a pixel is the one of the landmarks. If this is the case, we can extract the landmark pixels as shown in Figure 5.3. There are four red landmark objects in (a),

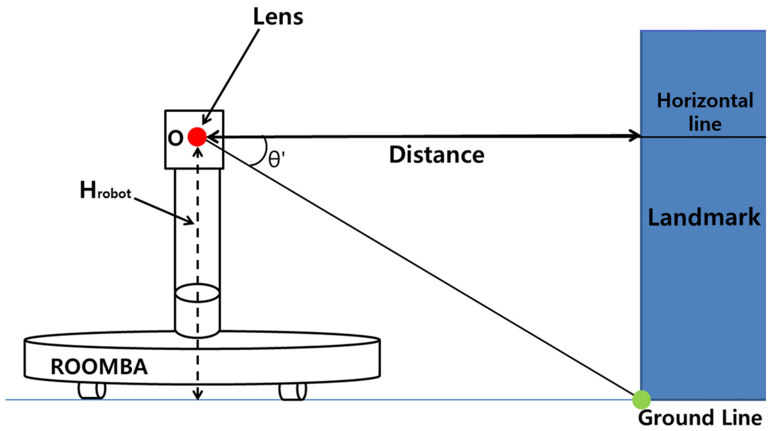


Figure 5.4: Principle of distance estimation. The mounted camera (O) has a height value (H_{robot}), and the red dot indicates the camera position. Then, the green dot of contact between the object (Landmark) and floor is equal to the one pixel on the ground line. The angle value (θ') between the horizontal line and ground line is directly proportional to the distance.

and we can form a binary image containing the landmark information (b) by using a simple HSV color filter. Thus, we can apply an adequate threshold value to change the current image to a binary image with sketchy information about the landmark. Then, there are only two types of pixels, black and white. Black pixels are pixels containing the landmark object, and the white represent the background. In Figure 5.3 (c), the panoramic form of (b), which is pixel-wise landmark information, is shown.

5.1.1.2 Distance estimation to collected landmark pixel set

After collecting the pixel-wise landmark from the image, we have to estimate the distance values. We obtain the panoramic image containing binarized landmark information. In one panoramic snapshot [720×90],as the horizontal x-axis indicates the angle, the unit length along the x axis is equal to 0.5 degree. The origin of the horizontal x axis is equivalent to the head angle at the moment of taking a snapshot, and its angular location can be changed from 0 to 359 degrees.

If there are no occlusions, we can see the all of the landmarks in this panoramic image at once. For example, in the ALV method (Lambrinos et al., 2000), panoramic images are used to find the direction of the landmark. Then, the upper and lower most parts of

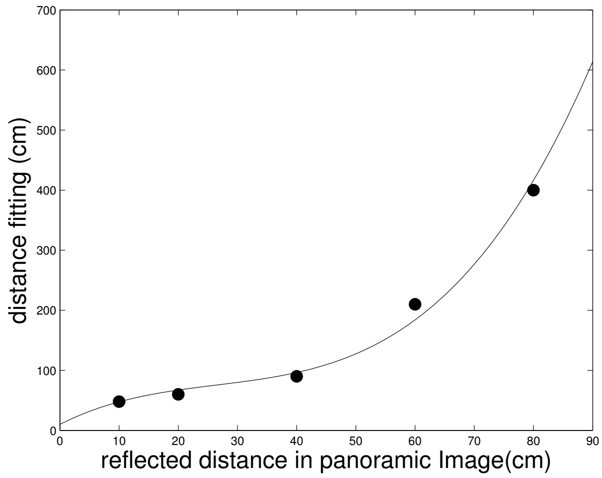


Figure 5.5: Graph of the relation formula for the distance-fitting equation.

the panoramic image are cropped away, and the horizontal information in the middle of the panoramic image is extracted before cutting. Along the lines of the ALV method, DID and ACV do not use distance information. Therefore, we tried to use the lower part of the panoramic image to obtain distance information. We can check that the angle (θ') between the horizontal line and ground line to the landmark is not directly proportional to the real distance to the landmark. Then, if we can find the relation between them, it can be used as the estimated distance value.

Figure 5.4 shows the principle of distance estimation based on the ground line of the image. The y-axis under the horizontal line of the panoramic image includes two types of pixels. The white one represents the background, and the black one, the landmark. See the example of the image in Figure 5.3. There are four landmarks. The nearest one has a short length from the bottom to the ground line, and the most distant one is longer than the short one. We can check that the distance in the picture is not identical to the real distance. To find the relation between the distance in reality and in the picture, we set some rules.

First, we treat the landmark pixel as noise when the size is very small. Before this, as we use only a binary process, the images contain a lot of noise. This rule thus reduces the error and improves the probability of finding the landmark. Second, we find the point of contact between the landmark and ground. From the bottom to the top of the panoramic image, the point of contact between the landmark and ground can be easily checked; no compensation is required for barrel distortion.

Based on these two rules, we can find the contact points and moderately remove the

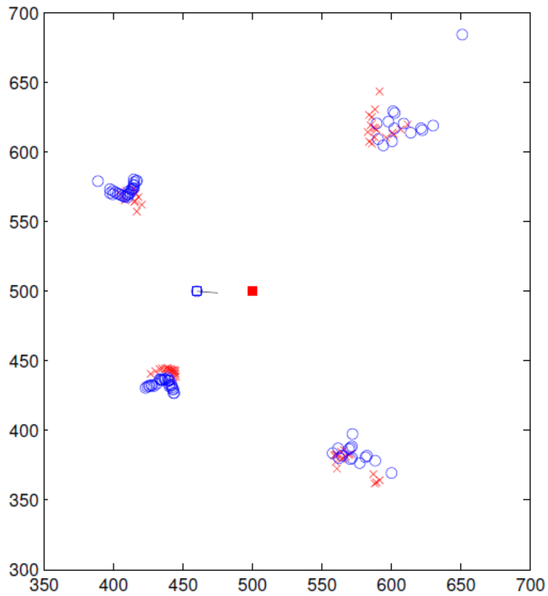


Figure 5.6: Example of the distance estimation from the panoramic Image. Distance fitting coefficient $a_1 = 5$, $a_2 = 0.143$, $a_3 = 0.0018$,$a_4 = 10$, red x is extracted landmark pixels from home cite (500,500), the blue o from the other cite at (460,500).

noise. We start checking the vertical data from the center of the robot (x = 0 pixel in the panoramic image). For a certain angle, we check the vertical data gradually and count the total number of landmark pixels. If the black pixels that represent landmark are found, and if their number exceeds the minimum landmark size, then the connected pixels in minimum continuity are checked. The minimum landmark size value and continuity value are manually determined by repeated experiments (the minimum landmark size is 10 pixels, and the minimum continuity is 3 pixels). If the conditions of landmark size and continuity are satisfied, then the first pixel is the contact point of the landmark and ground. Then, using the angle and distance in the image to this pixel as a reference, the direction and length of the landmark vector are calculated. After checking all of the pixels in the panoramic image, this angle is linked to real distance estimation.

We suppose that the relation between the real and reflected values can be written as a polynomial like

$$d = a_1 d_r^3 + a_2 d_r^2 + a_3 d_r + a_4 \tag{5.1}$$

$$a_1 = 5,\ a_2 = 0.0143,\ a_3 = 0.00018,\ a_4 = 10$$

where reflected distance (d_r) with polynomials and real distance (d). Because we use the numerical method to find the real distances from the reflected ones. Without specific physical modeling, we determine the relation by experiment for the distance variation, including defective instruments. Our objective is not to find a precise coefficient but a suitable one. Therefore, our outcome has some error. However, if we once obtain a rough pattern of landmark distribution, we can get the potential outcome using our algorithm. In Figure 5.6, we can see the matching between the two landmark sets. Despite the differences in each observation, the distance estimation from the reflected image was relatively successful. Thus, we can get both angular positions and estimated distances for each landmark pixel.

5.1.1.3 Homing navigation

We can extract both the angular location of pixel-wise landmarks and the estimated distances. Then we change these PLVs of two images into a potential function to find the home position. If our potential function(M) is positive convex, and the minimum point is at the home position, then the returning course following the gradient of the potential function ($-\nabla M$) must be reached at home (X_0). The gradient of Method 1 can be written as follows:

$$\overrightarrow{H} = -\nabla M = \overrightarrow{O(X)} - \overrightarrow{O(X_0)} \tag{5.2}$$

where $\overrightarrow{H}$ is homing vector which is equal to the gradient of potential field (M). It can be represented by subtraction of two summations of PLVs ($\overrightarrow{O(X)}$ and $\overrightarrow{O(X_0)}$) and $\overrightarrow{u}$ is unit vector pointing each direction (θ). Based on snapshot model, the one is from home position (X_0) and the other is from current position (X). Attached with modelling including pixel-wise landmark feature with estimated distances, it can be calculated like

$$\overrightarrow{O(X)} = \sum_{\theta} C_\theta r_\theta \overrightarrow{u}(\theta) = \sum_{i} \left(\frac{\eta R_i C_i}{||X_i - X||}\right)[a_i - x, b_i - y] \tag{5.3}$$

where θ is angle of holistic approach, i is for i-th landmark, C is feature value (color), R is radius of each object, η is normalize factor, r is distance between i-th landmark location ($X_i = (a_i, b_i)$) and position ($X = (x, y)$). Then we apply it into first function and gradient (M) can be changed into

$$\nabla M = \sum_i \left(\frac{\eta R_i C_i}{||X_i - X_0||}\right)[a_i - x_0, b_i - y_0] - \left(\frac{\eta R_i C_i}{||X_i - X||}\right)[a_i - x, b_i - y] \tag{5.4}$$

where home position ($X_0 = (x_0, y_0)$) and current position ($X = (x, y)$). Because two summation of PLV sets of overall angular positions with estimated distance values (r_0,r) can be shortened by cylinder shape approximation of landmarks that is equal to the one's of ALV method. Also, we check three terms which confirm unique convergence to home (X_0) position. Detail procedure is introduced in Appendix. Then mathematical expression about calculated homing vector ($\overrightarrow{H_t}$) with distance can be indicated by

$$\overrightarrow{H_t} = \sum_\theta \delta_0(\theta) C_0(\theta) r_0(\theta) \overrightarrow{u}(\theta) - \sum_\theta \delta(\theta) C(\theta) r(\theta) \overrightarrow{u}(\theta) \tag{5.5}$$

where δ_0 is binary object information of object at home position and δ at current position. It is the function of PLV selection that is 1 to selected pixel and 0 to the unselected. $C_0(\theta)$ and $r_0(\theta)$ are observed feature value and distance at home position. $C(\theta)$ and $r(\theta)$ at current position.

Finally, we obtain the direction to the desired position by using two PLVs from the two snapshots—one from the desired position and the other, the current position. If the robot uses our model, it can change the snapshot to PLVs at home. During the journey, at the current location, the robot takes a snapshot and also changes the data to PLV data. At each moment, the agent checks whether the current location is home by calculating the S function (simple image difference function). If the location is not home, the agent simply calculates the homing direction by using the PLV sets that are much lighter than images. On completing the journey, the robot reaches home.

This method does not use any complicated or heavy image-processing algorithm. Therefore, the speed is higher than that of other general methods, and the method is also accurate. However, as this method is based on pre-designated landmark object information and as the reliability of distance estimation is not perfect, we attempted to upgrade our algorithm to treat various objects without pre-knowledge.

The simulation for method 1 is described here. First, we designate the positions, colors, and radius of landmarks to build the simulation environment. Then, we collect the

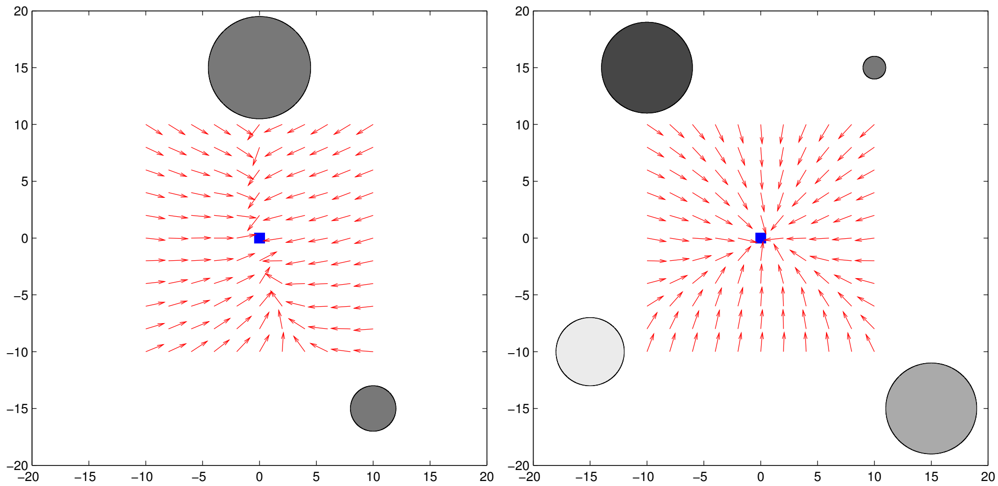


Figure 5.7: Simulation output in the vector map containing the homing vectors using method 1. Circles of various colors are landmark objects, and blue square indicates the home position. Red arrows indicate the calculated homing directions at each point.

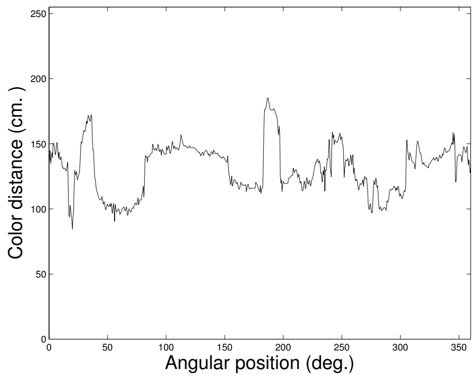
color pixels in a horizontal line, and we directly use the distance values that are not generated from the ground line. Therefore, there are no errors in measurement. We collect the landmark pixel by filtering the background based on a fixed color value and extract the PLVs. Then, the final homing direction is calculated by comparing the home PLV set and the current one. Figure 5.7 shows the outputs. Some objects are present around home, and we can calculate the homing direction at various points. For each environment, we can directly observe the pattern of homing convergence in each environment.

5.1.2 Navigation Using Pixel-wise Landmark Only Using Color Intensity

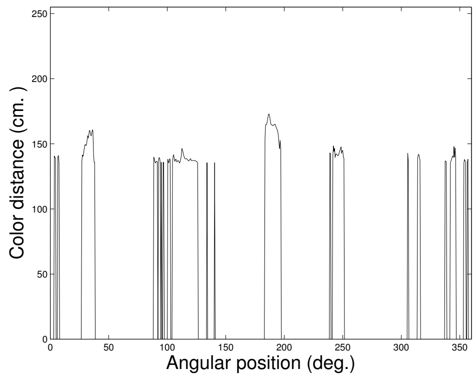
In the previous section, we presented the principle of the next algorithm modified to a normal situation without any object information. This involves four steps. First, we easily and simply pick the necessary pixel-wise landmarks of each image using color distribution. Second, we build the landmark vectors by using the extracted pixel-wise landmark information. Then, we can create the set of landmark vectors that can represent each position. Third, we find the expected homing direction by PLV matching to compare the home and current information. Fourth, we present adaptability to our method to change the model into a proper form according to the returning process. We

(a)

(b)



(c)



(d)

Figure 5.8: Example of landmark pixel extraction. (a)Input panorama image (is equal to the one in Figure 5.9), (b) extracted image using intensity sorting algorithm, (c)spectrum of the omni image, and (d) spectrum of the fitting image

present the detailed processes in the following sections.

5.1.2.1 Collecting landmark pixel

Unlike method 1, method 2 is a general method that does not need pre-knowledge about surrounding objects. Therefore, we selected the PLVs of each image. Earlier, we converted the omnidirectional image into a simple sequence of color values by image processing. As the landmark vector points to the landmark, it has both length and angle components. Therefore, we propose two procedures to generate the PLV without distance estimation.

First, we use all of the color sequence data as vector lengths. We call it as meta-distance, which acts as distance but is not equal to the real distance. There are 720 color data values (with 0.5-degree angular resolution). Then, we form 720 vectors using both the angular position of the pixel and the color value as the distance. In Figure 5.8 (a), the gray panorama image from the home position is shown. We pick 720 pixels in the

(a)

(b)

Figure 5.9: The example of image process. (a)input omni-directional image, (b) unfolded panoramic image (c) after using histogram equalization.

(a)

(b)

(c)

(d)

Figure 5.10: Example of automatic pixel-wise landmark extraction. The first one is the image at home position (50,50), and the third one is at a different position (54,58). The second and fourth are the PLV extracted versions of the first and third, respectively. The white region is the non-selected part. The number of pixels is automatically selected by using both histogram equalization and score calculation.

horizontal line and generate the sequence data with color and angular values. Then, we generate the spectrum of the sequence data in Figure 5.8 (c).

Second, we apply the sorting process to select useful vectors to build PLVs. Instead of using the threshold value, we sort the sequence of data according to the size of color value, and select the designated number of the data with higher values. If a pixel has high color value, then the order of the pixel data after sorting is very early, and it is picked up to build the landmark vector. Unlike the other two methods, in sorting, a fixed number of pixels are taken from the sequence of data. Therefore, we can pick an equal number of PLVs from the pixels. In Figure 5.8, the selected part obtained after this procedure is shown.

Further, we apply modifications to enable automatic operation. Automation can be re-

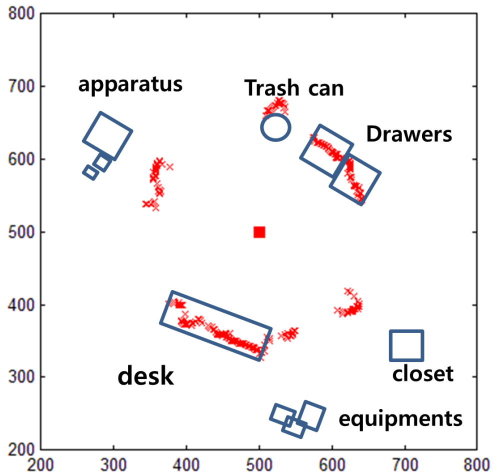


Figure 5.11: Top view of the environment shown in Figure 1. The various polygons represent the objects in the reference map, and x shows the landmark pixel after fitting 270 pixels

alized based on histogram equalization. It is a technique for adjusting image intensities, which is a normalization technique. In vision research, this technique is generally applied to reduce illumination errors. However, we use it for selection. We attempted to create an algorithm that automatically finds the optimal number of pixels between two images. There are three courses.

The first one is the normalization of two images by applying histogram equalization. Then, all the pixels are sorted from 0 to 1. The second one sorts the normalized output. Thereafter, based on normalized home data, we check how well all the pixels of the current one match the home pixels. In this course, we calculate the score according to the number of pixels that changes from 50 (minimum) to 720 (maximum). At each time, the score calculated by the sum of the top three components of color matched the list that has absolute angular differences between the current pixel with all the pixels in the home data arranged in the ascending order of color differences. In this case, the angular differences must be less than 180 degrees. In short, it roughly (picks top 3 units) finds the well-matched cases of pixels with small location changes (under 180 degrees). The final score is calculated by dividing the counted score with the square of the number of pixels. Then highest value is output. Figure 5.10 shows an example of automatic PLV selection. The output has small errors, but our algorithm can operate with these errors.

In Figure 5.11, we show the other example distribution of PLVs obtained by the selecting method. The red square indicates the home, and the x-shape indicates the end point of the landmark vectors from the home position. We indicate the real arrangement of the landmark in various polygons with name tags. We can observe the effect of the sorting process that successfully extracts the landmark, and we also verify that the color distance is not equal to the real one.

In this part, we generate the landmark vectors using various methods. Unlike other traditional methods, we do not use the real distance to the landmark or the exact angular position of the landmark in the images. However, we apply some rules, and we pick some pixels that satisfy the rule. Then, we generate the landmark vectors that have color value as the distance. There are 720 pixels for one omnidirectional image, and after fitting, there are about 90 to 500 landmark vectors. We reduce the 640 $\times$ 480 omnidirectional images into proper PLVs by approximately 90 to 300 times. Then, in the next part, we compare them and find the way to reach home.

5.1.2.2 Homing Navigation

We generate pixel-wise landmark vector sets that fit into navigation by using color values. Our object modeling is applied here, and the final sum of PLVs with color values (C_i) can be expressed as follows:

$$\overrightarrow{O(X)} = \sum_{\theta} C(\theta)\overrightarrow{u}(\theta) = \sum_{i} \left(\frac{\eta R_i C_i}{||X_i - X||}\right) \cdot \frac{1}{||X_i - X||}[a_i - x, b_i - y] \tag{5.6}$$

where θ is is the angular position of a landmark. C_i is a feature value (color) and R_i is radius of the i-th landmark given at location ($X_i = (a_i, b_i)$) from the current view at $X = (x, y)$. As our model does not have any distance information, there is one more term ($1/r_i$, $r_i = ||X_i - X||$).

Based on the sum of PLVs without distance information, we test a gradient of potential field like the prior method (5.4) with distance information as follows:

$$\overrightarrow{H} = -\nabla M = \overrightarrow{O(X)} - \overrightarrow{O(X_0)} = (X_F, Y_F) \tag{5.7}$$

where homing vector can be calculated as negative gradient of potential field and ∇M is a gradient of potential field (see the potential M in appendix). Unlike method 1, a simple gradient of the potential field cannot guide homing because the convergence conditions are not satisfied. The convergence conditions are given by

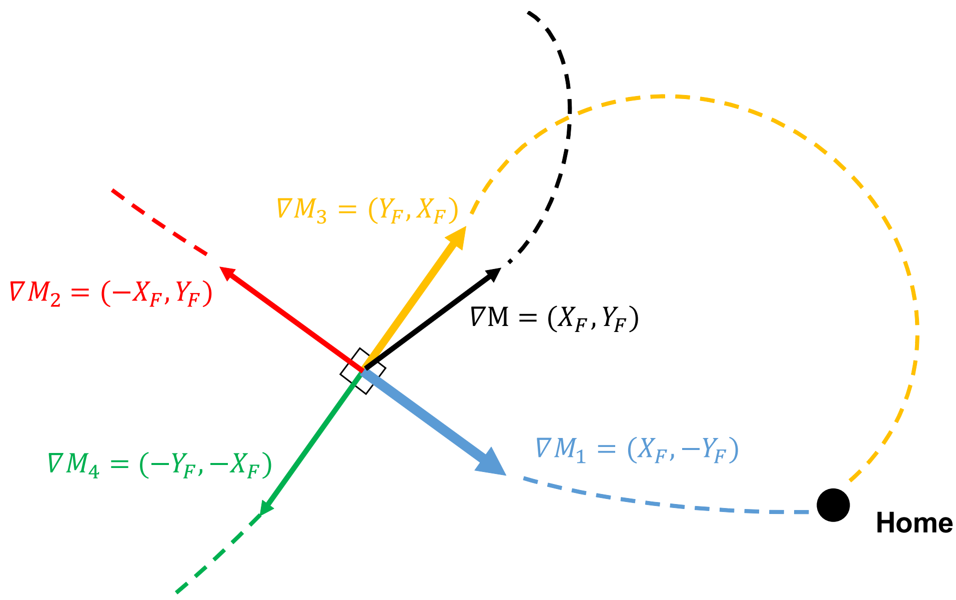


Figure 5.12: Concept of route selection in our model without distance information. ∇M is equal to the one using distance information but it fails. ∇M_1 and ∇M_3 shows successful homing using variation with different patterns. ∇M_2 is reverse of ∇M_1 and ∇M_4 is reverse of ∇M_3.

$$\det(J(\nabla M(X))) > 0 \quad M_{xx}(X) > 0 \quad \nabla M(X_0) = 0 \tag{5.8}$$

where

$$J(\nabla M) = \begin{bmatrix} \frac{d^2M}{dx^2} & \frac{d^2M}{dxdy} \\ \frac{d^2M}{dxdy} & \frac{d^2M}{dy^2} \end{bmatrix} = \begin{bmatrix} M_{xx} & M_{xy} \\ M_{yx} & M_{yy} \end{bmatrix} \tag{5.9}$$

This Jacobian matrix can be calculated by quadratic differential of each term including M_{xx} for quadratic differential about x and M_{yy} about y. The above potential M does not satisfy one of convergence conditions, $det(J(\nabla M)) > 0$, and cannot guide homing. Thus, new gradient of potential fields to support the convergence are suggested. We propose four homing vectors $(\nabla M_1, \nabla M_2, \nabla M_3, \nabla M_4)$ as given by:

$$\begin{aligned} \nabla M_1 &= -\nabla M_2 = (X_F, -Y_F) \\ \nabla M_2 &= -\nabla M_1 = (-X_F, Y_F) \\ \nabla M_3 &= -\nabla M_4 = (Y_F, X_F) \\ \nabla M_4 &= -\nabla M_3 = (-Y_F, -X_F) \end{aligned} \tag{5.10}$$

where ∇M_1, ∇M_2, ∇M_3 and ∇M_4 are extracted from ∇M which is not successful without distance information. These terms do not use simple X_F and Y_F; instead, they change their sign or order. Figure 5.12 shows the concept of these four routes. Black one (∇M) using (X_F, Y_F) fails in homing but there are two variations (∇M_1 and ∇M_3) show successful homing routes. A sequence of a homing vector produces a homing route what homing vectors guide homing should be investigated. Then those four gradient functions are given by

$$
\begin{aligned}
\nabla M_1 &= \sum_i \left(\frac{\eta R_i C_i}{||X_i - X_0||^2}\right)[a_i - x_0, -b_i + y_0] - \left(\frac{\eta R_i C_i}{||X_i - X||^2}\right)[a_i - x, -b_i + y] \\
\nabla M_2 &= \sum_i \left(\frac{\eta R_i C_i}{||X_i - X_0||^2}\right)[-a_i + x_0, b_i - y_0] - \left(\frac{\eta R_i C_i}{||X_i - X||^2}\right)[-a_i + x, b_i - y] \\
\nabla M_3 &= \sum_i \left(\frac{\eta R_i C_i}{||X_i - X_0||^2}\right)[b_i - y_0, a_i - x_0] - \left(\frac{\eta R_i C_i}{||X_i - X||^2}\right)[b_i - y, a_i - x] \\
\nabla M_4 &= \sum_i \left(\frac{\eta R_i C_i}{||X_i - X_0||^2}\right)[-b_i + y_0, -a_i + x_0] - \left(\frac{\eta R_i C_i}{||X_i - X||^2}\right)[-b_i + y, -a_i + x]
\end{aligned}
\tag{5.11}
$$

where C_i is feature value (color) and R_i is radius of the i-th landmark at location ($X_i = (a_i, b_i)$) for the current location ($X = (x, y)$) and the home location ($X_0 = (x_0, y_0)$). Four different types of suggested models are perpendicular to each other, and two different directions are opposite ($\nabla M_1 \cdot \nabla M_3 = 0, \nabla M_1 = -\nabla M_2, \nabla M_3 = -\nabla M_4$). We observe the convergence conditions of these four models. According to Equation (5.8), the convergence condition for each gradient can be written as

$$
\begin{aligned}
\nabla M_1(X_0) = 0, det(J(\nabla M_1(X))) \geq 0, (M_1)_{xx} &= \sum_i -\eta R_i C_i \cos(2\theta_i)/D_i^2 = K_1 \\
\nabla M_2(X_0) = 0, det(J(\nabla M_2(X))) \geq 0, (M_2)_{xx} &= \sum_i \eta R_i C_i \cos(2\theta_i)/D_i^2 = -K_1 \\
\nabla M_3(X_0) = 0, det(J(\nabla M_3(X))) \geq 0, (M_3)_{xx} &= \sum_i -\eta R_i C_i \sin(2\theta_i)/D_i^2 = K_2 \\
\nabla M_4(X_0) = 0, det(J(\nabla M_4(X))) \geq 0, (M_4)_{xx} &= \sum_i \eta R_i C_i \sin(2\theta_i)/D_i^2 = -K_2
\end{aligned}
\tag{5.12}
$$

where we set K_1 and K_2 for simplicity. These four cases like $(M_k)_{xx}$ are quadratic differentials about x and they can be represented by two constants(K_1 and K_2). Hence, at most two ($K_1 \neq 0$ and $K_2 \neq 0$) and at least one of four suggested models will have homing convergence.

Among three convergence conditions, ($\nabla M_k(X_0) = 0, det(J(\nabla M_k(X))) > 0, M_{k,xx} > 0$ for $k = 1 \sim 4$), one condition $\nabla M_k(X_0) = 0$ is satisfied. The above four models are analyzed in details to check the convergence.

Case 1. $K_1 \neq 0$ **and** $K_2 \neq 0$ **:** First, if these two constants are not zero, Equation (5.12) can have two convergence models. (see appendix).

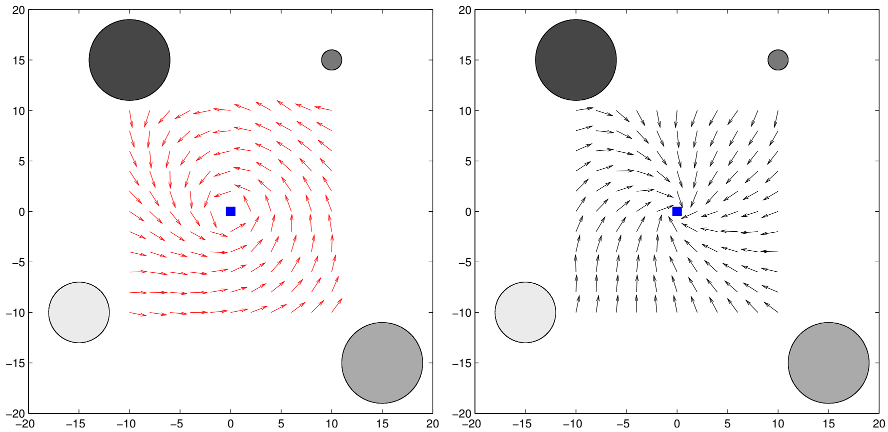


Figure 5.13: Simulation for method 2. Blue square refers to home position, and there are four landmarks with different size, location, and color information. Left one uses M_1 and right one uses M_3

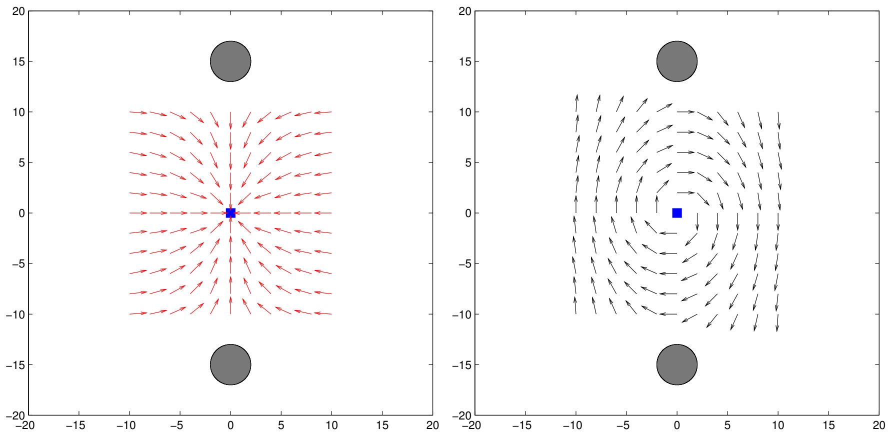


Figure 5.14: Simulation of method 2 under a special condition ($K_1 > 0, K_2 = 0$). There are two identical landmarks, but we control the location to obtain this condition. The left one corresponds to M_1 and right one to M_3.

Case 2. $K_1 \neq 0, K_2 = 0$ **or** $K_1 = 0, K_2 \neq 0$ **:** With this condition, two of four models have $(M_k)_{xx} = 0$. At least one of models will have convergence with non-zero K_1 or K_2. For example, the case with $K_1 = 0$ and $K_2 > 0$ has ∇M_3 as solution.

Case 3. $K_1 = 0$ **and** $K_2 = 0$ **:** With this conditions, there is no convergence model violating $(M_k)_{xx} > 0$.

Simulation was also conducted for method 2. We design various simulation environments, and we collect the color pixel information for landmarks. As there are four models, we use different colors for plotting the homing directions of each model by Equation (12) (∇M_k). Here, ∇M_1 is indicated in red; ∇M_2, in blue; ∇M_3, in black; and

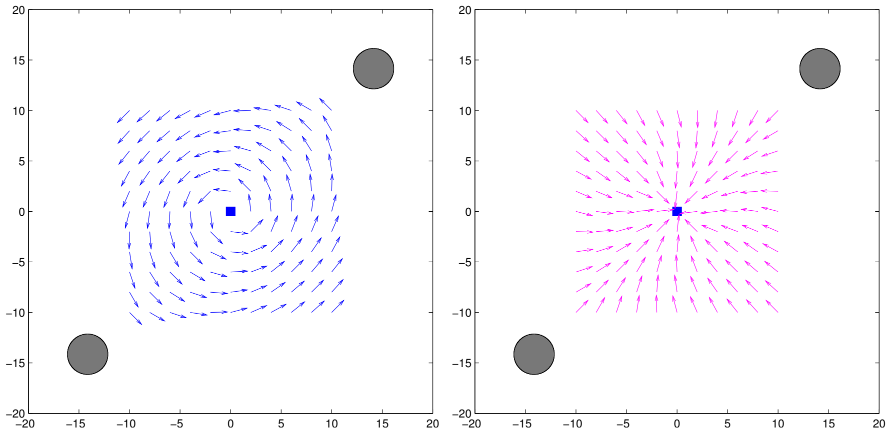


Figure 5.15: Simulation of method 2 under a special condition ($K_1 = 0, K_2 \neq 0$) with different arrangement. There are two identical landmarks, but they differ from the one in Figure 5.14. The left one corresponds to M_2, and the right one, to M_4.

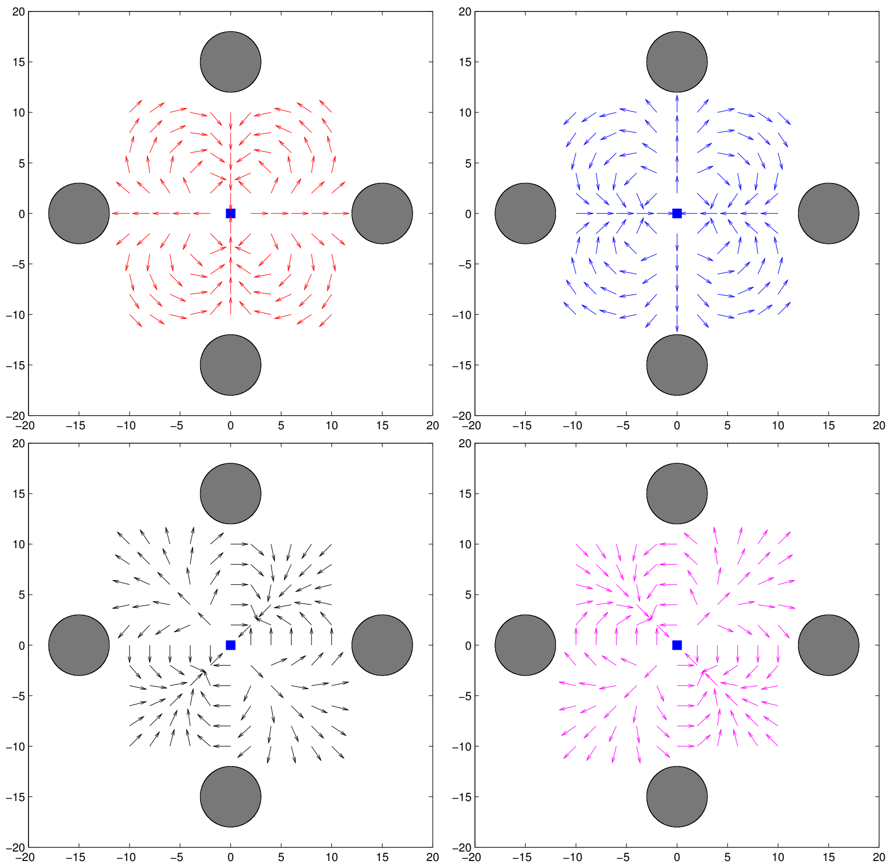


Figure 5.16: Simulation of method 2 under the most special condition ($M_{xx} = 0$ and $D(X_0) = 0$). There are four identical landmarks and arrangements. Four vector maps show the different reactions of these four models.

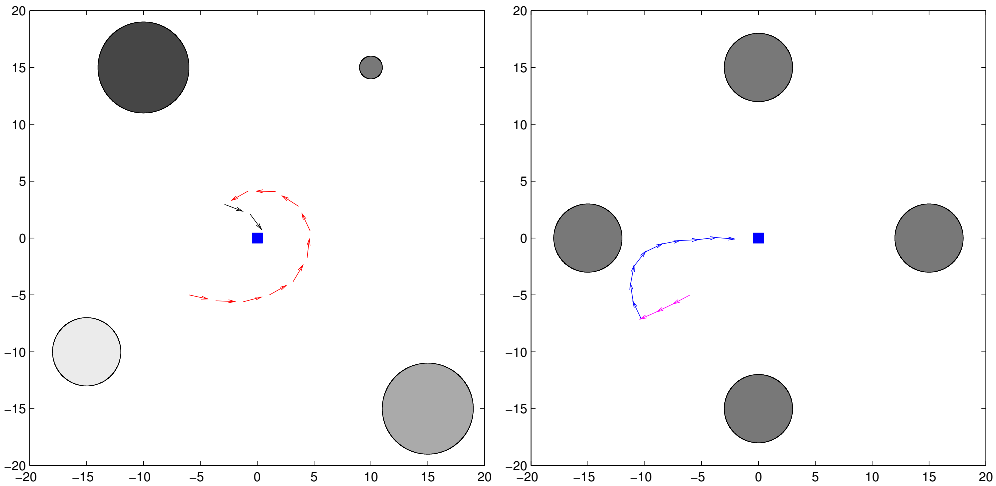


Figure 5.17: Example of homing navigation using a combination of models. The left one changes the model from M_1 to M_3 in the normal situation. The right one changes the model from M_4 to M_2 in the special situation. The starting point is (-5,-5), and the home position is (0,0)

∇M_4, in magenta.

Figure 5.13 shows the results for $K_1 \neq 0$, $K_2 \neq 0$. The left one uses M_1, and the right one uses M_3. The other two models have opposite homing directions. These two figures have different homing route patterns. Figure 5.14 and 5.15, two situations show the homing results for either $K_2 = 0$ or $K_1 = 0$. These environments are generated to test the solution for these special situations. However, both cases have positive $det(J(M_k(X)))$. There must be one solution among four models. In Figure 5.14, M_1 can be that solution, and M_2 has the opposite homing pattern since $K_2 = 0$. In addition, M_3 and M_4 shows incomplete homing patterns. Figure 5.15 shows similar tendency with $K_2 = 0$. Figure 5.16 shows the results for $K_1 = K_2 = 0$. In this case, $(M_k)_{xx} = 0$ and $det(J(\nabla M_k) = 0$. There are two regions for successful homing or failure. M_1 shows success routes along y-axis, but failure along the x-axis. All the points along the x-axis have $(M_k)_{xx} = 0$. Similar vector patterns are observed in the other cases.

We test the homing simulation using a combination of the four models. With home snapshot, it randomly selects one model and follows the answer. We performed image differences between home and current image. If this model is not appropriate (increasing differences), then it changes model instead of current one. Examples are shown in Figure 5.17. Our agent can arrive at home in both cases without referring to the condition of the environment.

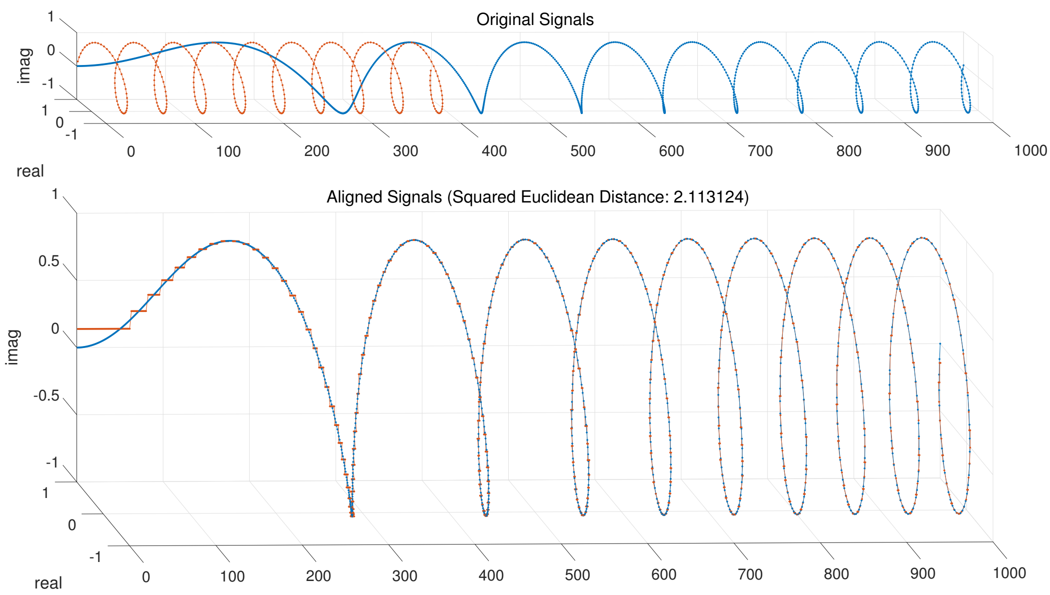


Figure 5.18: Dynamic Time Warping (DTW) : The blue signal is the original signal and the orange one is warped in the time domain. Then an Aligned output signal is produced (the second signal) by using squared Euclidean distance.

Finally, we find the homing direction by using both home and current omnidirectional images into two methods. Unlike other researches, in this study, we use a new concept of pixel-wise landmarks and theoretically prove our model. We describe both the simulation and experiment outputs in the next section.

5.1.3 Navigation Using Pixel-wise Landmark Matching Based on DTW

5.1.3.1 Dynamic Time Warping

In time series analysis, dynamic time warping (DTW) is one algorithm used for measuring similarity between two temporal sequences. There are various sequences that are compatible with DTW, including audio, video, graphics, routes of gestures, and any data which can be converted into a linear sequence A well-known example is automatic speech recognition, which tries to solve the problem of different speaking speeds and amplitudes (Sakoe and Chiba, 1978). DTW can also be used in partial shape matching such as macro gesture recognition.

Note that there are some restrictions when applying DTW. The sequences are 'warped' non-linearly in the time dimension. Also, the start and end points of both signals essentially must be equal. Thus, DTW is a method that calculates an optimal match between two given sequences under those restrictions.

In the calculation process, we can arrange the two sequences with an unknown sequence on the x-axis, and the base template one on the y-axis. Both sequences start on the origin of the grid. Each cell of the grid has a calculated distance between the two sequences. To find the best match between them, we find a path that minimizes the total distance between them. In this part, based on restriction, start point must be on the origin of the grid so that the minimum distance summation on each cell is calculated by comparison. The comparison process must proceed in the direction of increasing x and y indices, called insertion, deletion, and match. At the end of the grid, once an overall best path has been found, the total distance between the two sequences can be calculated. This procedure is designed to find all possible routes and it is apparent that for any reasonably sized sequences under restrictions, the number of possible paths through the grid will be very large. Therefore, overall calculations require extensive computation. There are alternative methods, such as SparseDTW (Al-Naymat et al., 2009), MultiscaleDTW (Müller et al., 2006),and Fast DTW (Silva and Batista, 2016), but we use the original version to obtain high accuracy.

5.1.3.2 Searching for Stationary Feature

The warping method based on holistic matching counts all of the errors resulting from pixel matching but requires extensive computation. We design a flow template modeled after Figure 2.9. These two cases show the pixel movements as flow according to horizontal and vertical movements with equal distance assumption. From previous research using complex feature point matching, we note the difficulties with matching these points at particular angles having large variation. A robust search requires complex matching algorithms like SIFT or SURF, which do not suit our aim of reducing computation time and complexity.

Our idea is to use the unchangeable points, and there are only two such points according to generated flow template. One is the convergence point that is behind the movement direction and the other is the diffusion point in front. These stationary features must be located at opposite directions and have similar intensity values. Also, accurate extraction of the line connecting the two stationary points is directly related

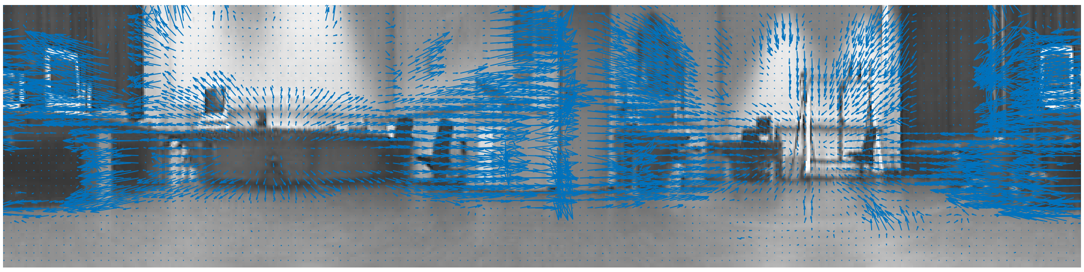


Figure 5.19: The example of extracted optical flows generated by the movements around two points (convergence and diffusion point) and marked as blue arrows.

to finding the homing direction. Therefore we try to find the location of this stationary feature point for pre-process of finding accurate homing direction.

5.1.3.3 Searching for Stationary Feature based on color distribution in horizontal line

First, based on the Figure 5.19, we design a new model for finding the stationary feature location. Because we think that the pattern around these two points can be found also in feature matching. Also, the optical flow is one of the bio-inspired visual feature matching which is widely used in extracting visual odometry information. Therefore we apply optical flow matching algorithm (Farnebäck, 2003) based on Lucas-Kanade method (Lucas et al., 1981). It calculates optical flow between two images by using quadratic polynomials between all the points, not corners, to find optical flow. Then the output shows robustness.

In this part, we have focus on only pixels around horizontal line to build simple method. Figure 5.20 shows the basic concept of those two points. We denote the two important points as two stationary features (CP: convergence point, DP: diffusion point). First we make a moving window having only horizontal shifted direction according to warping model. Next, we use correlation between direction of feature output and designing moving window.Then we calculate the distribution of correlated output for each direction and find the maximum one as optimal direction.

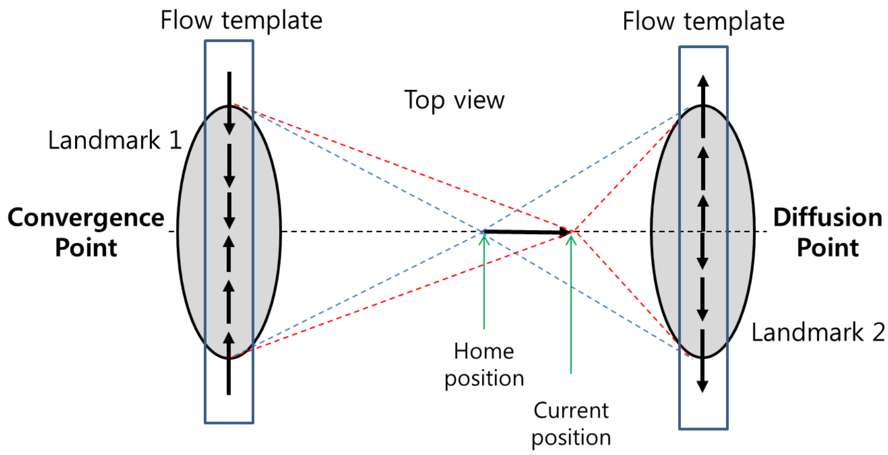


Figure 5.20: The idea of matching based on two important points. The gray circle means surrounding objects. When the agent moves from home position to current position, then flows are generated by the movements around two points (convergence and diffusion point)

5.1.3.4 Searching for Stationary Feature based on Entire Image Pixels

In this part, we suggest another one based on optical flow matching. Different to first method, we use all part of the 2 dimensional input image which is compared with reference image.

Figure 5.21 shows the principle of finding stationary points based on optical flow. First one shows the expectation of pixel movement around those two stationary points that are located at opposite direction to each other.

Then, similar with first method, we try to find the specific location of stationary point from the matched output. We use correlation between direction of feature output and designing moving window. Also, this moving window includes two regions for extraction. And these region have center converged and diffused direction from each center. Next, we calculate the distribution of correlated output for each direction and find the maximum one as optimal direction.

As mentioned above, our alignment method based on a visual compass has some errors associated with an agent's position when located far from the home location. We add one more step to compensate for non-aligned input in this case. The idea is to adjust the search area by swinging those matches in an attempt to better align with the home

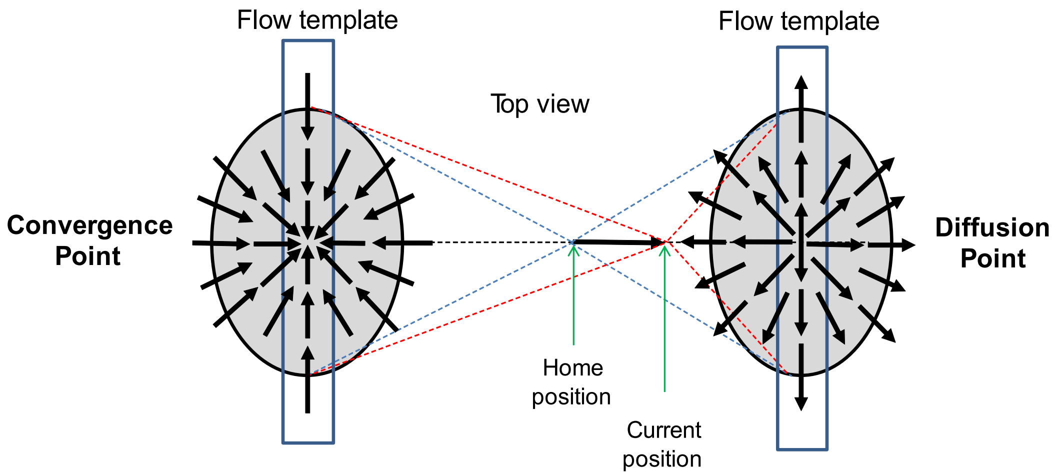


Figure 5.21: The idea of matching based on two important points based on optical flow. The gray circle means surrounding objects. When the agent moves from home position to current position, then flows are generated by the movements around two points (convergence and diffusion point).

location. For example, the previous steps only compare angular locations between two images. However, the inputs with aligned error (distant locations) are repeatedly tested with small rotated variations of the current view to find a more accurate location with respect to the home location.

5.1.3.5 Matching based on Dynamic Time Warping

In this section, we attach the dynamic time warping algorithm. As mentioned in the Introduction, the DTW method has some restrictions. First, the sequences are warped non-linearly in the time dimension. Also, start and end points of both signals are basically fitted. Through the above steps, we find the two stationary points which have fixed angular locations different to other pixels. These calculated points satisfy the condition of both start and end points needed to apply DTW. Based on the diffusion point as the start location, extracted wing data have warped angular locations in relation to each other. Therefore we consider those warped angular locations as time-shifted warping and we apply the original DTW to our navigation model.

In the calculation process, we arrange the two sequences into a distance matrix, one with unknown sequence on the x-axis and the base template on the y-axis. This unknown sequence is the current view and the base template is from the home snapshot.

Both sequences start on the origin of the grid of the matrix. Each cell of the grid has a calculated distance between two indexes of sequences. The algorithm then tries to find the best match route between them, and we can find a path that minimizes the total summation of distances between the two sequences. Based on DTW restrictions, the start point must be on the origin of the grid and it is the diffused point, which is one of stationary points. Next, DTW calculates minimum distance summation on each cell by a comparison process. The comparison process proceeds in the direction of increasing x and y indexes, called insertion (only y is increased), or deletion (only x is increased) and match (both x and y are increased). After reaching the end of the grid, it can find the best route by back tracking from this end point to the start point. This back tracking checks all possible routes having various distance summation values, which represents a very large number of possible paths requiring extensive calculations. That is, if we perform a full DTW search of all combinations between two images, then the output can be accurate but the number of calculations required is too large to use in real time. Instead, we can apply DTW in our model by attaching the matched stationary points as the start point. Then the combination of DTW and stationary point matching reduces computation time and makes our model robust.

Figure 5.22 shows examples of matched routes. The blue line is the case using two images having small differences. Red and black lines indicate current view images that have large differences and locations in opposite directions from the home snapshot image. In short, we can determine which combination is effective for navigation purposes.

Based on the above results, we have divided the combination method into two broad categories

5.1.3.6 Wing matching method

Based on the snapshot model, a snapshot image from the home position is compared to the current view. There must be two stationary points which can be calculated using the methods previously described. We use both wing data and two stationary points in this method. This first method is called the wing matching method and it uses two stationary points, the diffusion point as the start point and the convergence point as the end point, and applies DTW matching between both wings' data.

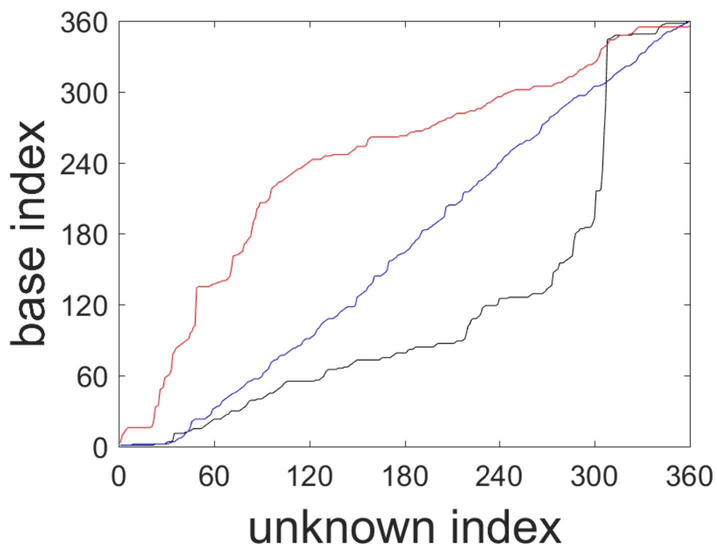


Figure 5.22: Three examples of route matching. The index of y-axis is called the base index (the home snapshot image). The index of the x-axis is called the unknown index (current view). The blue line is the matched output between left wings (L_θ) of the snapshot image (located at (5,8) of Vardy set)) and an example image from very near to the snapshot (location (4,8)). The black line is between the snapshot and a distant image (5,14). Red line is between the snapshot and another distant image (5,1).

The second part of Figure 5.23 shows the case using wing matching. In this figure, two stationary points are selected. The green one is the diffusion point and the red one is the convergence point. The color distribution of each wing is calculated and two wings are matched (right and left wings).

5.1.3.7 Unifying matching method

If our stationary point extraction is perfect, then DTW matching is theoretically perfect. But our current alignment algorithm with visual compass has errors, especially in distant locations. Then having a false aligned environment can produce errors by using wing matching. Therefore we developed a version called unifying matching which improves the DTW calculation by unifying the two wings.

In this version, we accept the instability of both alignment and extraction of stationary points. Therefore only one diffusion point is used as the start point and the data size for DTW matching is doubled by unifying the two wings into one. If the data length is L, then the length of each wing is L/2 and the overall number of DTW calculations for

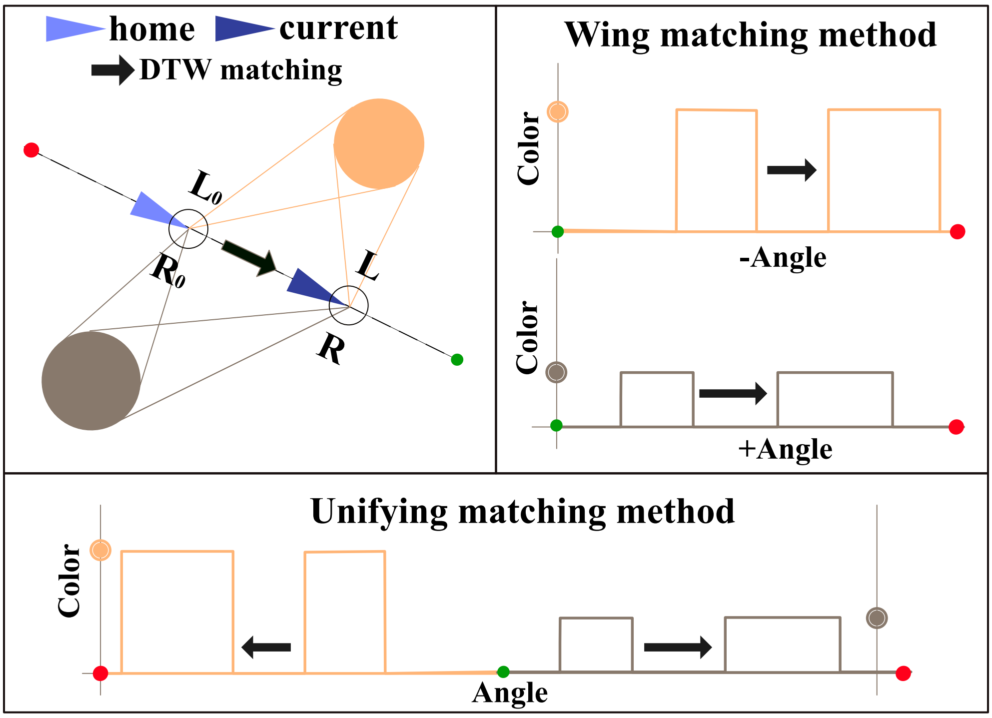


Figure 5.23: The concept of matching algorithm including score information. The first figure shows an example environment including two landmarks. One is bright orange color and the other is dark brown. Home and current location are marked by colored triangles. Green (diffusion point) and red (convergence point) dots are aligned. Final calculated homing direction by DTW matching is marked as a black arrow. The second figure shows the output by using the wing matching method. The diffusion point is selected as the start point and the convergence point as the end point of DTW matching. Upper graph shows the information of Left wing (L and L_0) in counter clockwise direction. Lower graph shows the information of right wing (R and R_0) in clockwise direction. The third figure shows an example using the unifying matching method. It uses the diffusion point as both start and end points. It connected both wings with double lengths and was directly applied into DTW without a convergence point. The circles on the y-axis of the graphs represent the information of color distribution in bar-code.

wing matching is $2 \times L^2/4$ compared to the unifying method, which is L^2. Thus, the unifying matching method allows the DTW output to be sufficiently accurate to avoid serious mismatches.

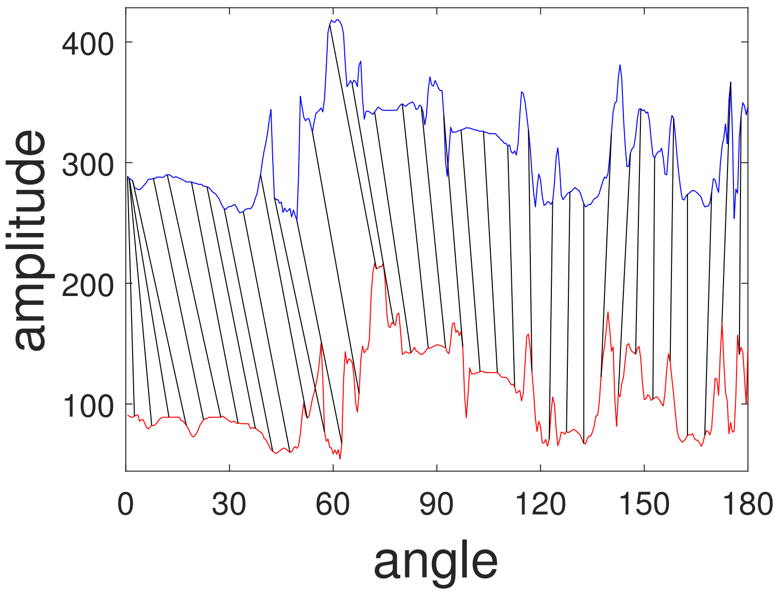


Figure 5.24: Example of connected-edge after inhibition. Blue line is for the data distribution of the current view and red is for the snapshot image. For figure, height of the blue line is increased to show connection. Each black line shows connected-edges for every 10 indexes.

5.1.3.8 Connected-edge inhibition

After matching, our method can generate matched information including some connections between two indexes. Each connection is represented as an edge and each edge has a start index from the current view and a target index from the snapshot (home) image. Based on DTW matching, there must be multiple start indexes for one target index. Also, there are multiple target indexes for one start index. Our goal is to inhibit these multiple choices. Based on the snapshot model, we make multiple start points be unique and select the one having the smallest distance. Therefore, we collect multiple start points from each target index and compare distances between them and inhibit other connected ones except for the one with the smallest distance. Then the number of connections is equal to the number of target indexes in the snapshot image. Figure 5.24 shows an example of output after inhibition.

Also, there are various choices for edge inhibition. First is the non-inhibition case that uses all of the connections in navigation. Second is the reverse inhibition case that reduces only target indexes. Third is the all inhibition case. We tested all, and found that the three cases have lower performances than our method.

5.1.3.9 Vector set comparison

Based on both snapshot model and DTW matching with inhibition, we have pixels of home and matched pixels of current view. We then create landmark vectors to find the homing direction. As we cannot know the distances of the pixels, we create unit length vectors pointing at the angular locations of snapshot pixels.

$$\overrightarrow{u}(\theta) = (\cos\theta, \sin\theta) \quad \sum_{\theta=-\pi}^{\pi} \overrightarrow{u}(\theta) = 0 \tag{5.13}$$

where $\overrightarrow{u}(\theta)$ is unit vector pointing to each angle (θ) and overall summation is equal to be zero. However, based on pixel-wise landmark matching, matched output can be presented like

$$DTW_f(\theta) = (\cos\theta', \sin\theta') \quad \sum_{\theta=-\pi}^{\pi} DTW_f(\theta) \neq 0 \tag{5.14}$$

where $DTW_f(\theta)$ is matching information of each angle then the summation is not zero like ACV. Each landmark vector has unit length and the summation of all pixels in the image is considered to be zero. Different to landmark vectors of snapshot images, landmark vectors of matched pixels ($DTW_f(\theta)$) are also generated with asymmetric properties. Then we can calculate the homing vector by summation of the differences between two asymmetric vectors. We suggest two methods. The one is wing-matching ($\overrightarrow{H}_{wing}$) and the other is unifying-matching ($\overrightarrow{H}_{unifying}$). These can be presented like

$$\overrightarrow{H}_{wing} = \sum_{\theta=-\pi}^{0} (\overrightarrow{u}(\theta) - DTW_{fL}(\overrightarrow{u}(\theta))) + \sum_{\theta=0}^{\pi} (\overrightarrow{u}(\theta) - DTW_{fR}(\overrightarrow{u}(\theta))) \tag{5.15}$$

$$\overrightarrow{H}_{unifying} = \sum_{\theta=-\pi}^{\pi} (\overrightarrow{u}(\theta) - DTWf_U(\overrightarrow{u}(\theta))) \tag{5.16}$$

where DTW_{fL} is pixel matching for left wing, DTW_{fR} for right wing and DTW_{fU} for unifying model. Then the overall calculation of the homing vector in wing matching is divided into two sets. On the other hand, the unifying matching method output ($\overrightarrow{H}_{unifying}$) uses a set of matching vectors. Then we can find the homing direction by comparing landmark vectors based on matched outputs.

5.1.3.10 Homing navigation

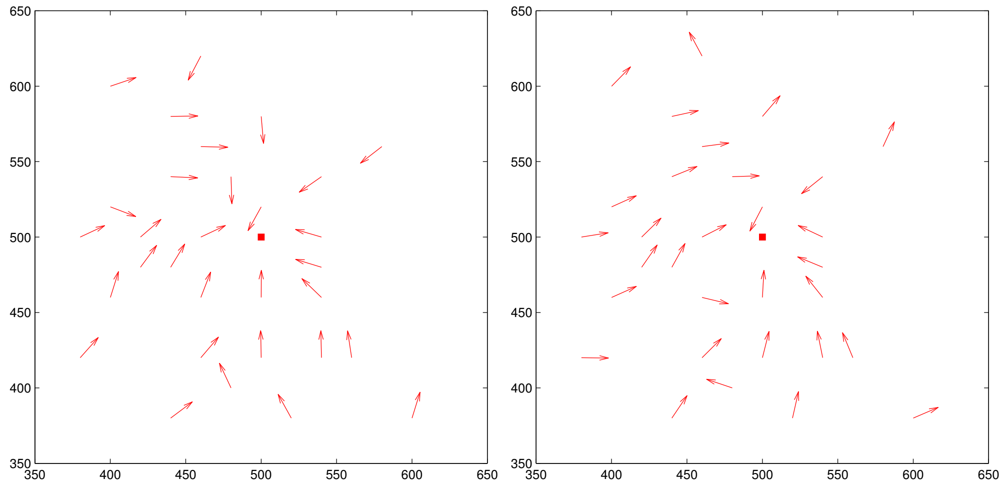
Finally, we created a new model using two important points related to stationary characteristics on a flow template. Based on the snapshot model, we align two images based on the visual compass. Next, we try to find two stationary points between two images based on correlation about flow direction. It calculates the differences between opposite pixels, and pixel flow of remaining pixels around focus angle. Then a particular angle is selected as the start point for DTW matching. This part focuses on reducing the overall calculation by using those points instead of full DTW matching.

However, the calculated homing vector has wrong information for the distance factor, because the length of the landmark vector is not equal to the distance. Therefore, we cannot use this homing vector and we have to update it. So, we apply homing sequences including multiple updates. The agent using our model has to find the homing direction at a current point and move small distances along this vector. Next, the agent collects another image after movement and updates the current view to update the homing vector. With repetition of a prior course, the agent can arrive at the home point. At each iteration, it tries to judge whether this point is home or not by calculating overall differences between the home snapshot and the updated current view.

5.2 Experiments and Results

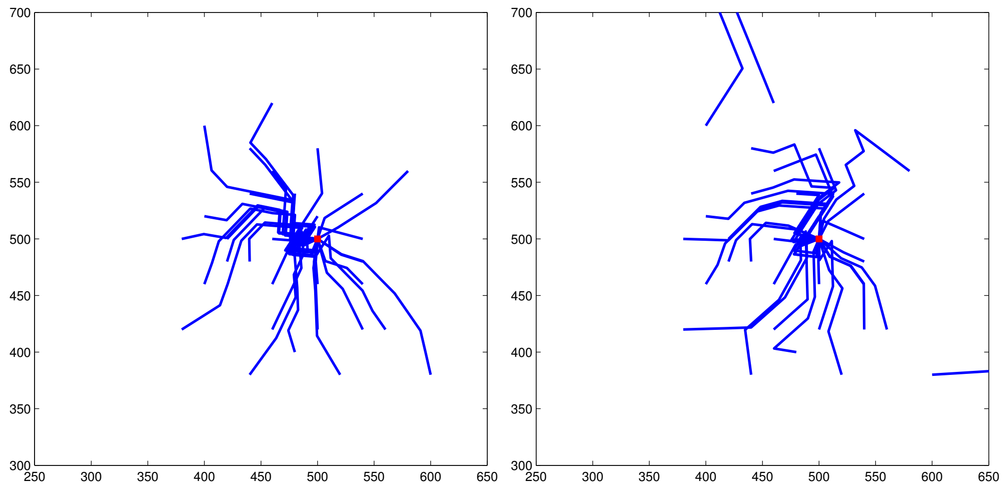
5.2.1 Experiments Using Pixelwise Landmark

Now, the robot experiment for method using distance information is described. The environment of this experiment is identical to the one shown in Figure 5.3. There are four landmarks in red color. Figure 5.25 shows the calculation output of homing direction at various locations. The first one is obtained using manually aligned images, and the second one is obtained using a visual compass algorithm for alignment. Despite the measurement errors, the homing convergence pattern is obtained. Figure 5.26 checks the homing routes marked by a blue line. The line pointing outward without a compass sensor indicate the fails; most cases are 100% in aligned condition and 82% in visual compass case. In addition, these failures occurred at an outer location far from the home position.



(a) (b)

Figure 5.25: The robot experiment output in a vector map including the homing vectors using method 1 a) with aligned images, and b) using visual compass.



(a) (b)

Figure 5.26: Robot experiment output in the homing route using method 1 a) with alignment, and b) using a visual compass

As mentioned above, method using distance information has two limits. One is landmark extraction using pre-knowledge about objects. This procedure is not perfect, and the output has obtrusive errors in spite of additional filtering processes. The other is the reliability of distance estimation based on the ground line. Both extraction of the ground line and distance estimation model have some errors. Then, the final output also has errors caused by those error factors, and we conduct additional experiments for second method without these procedures.

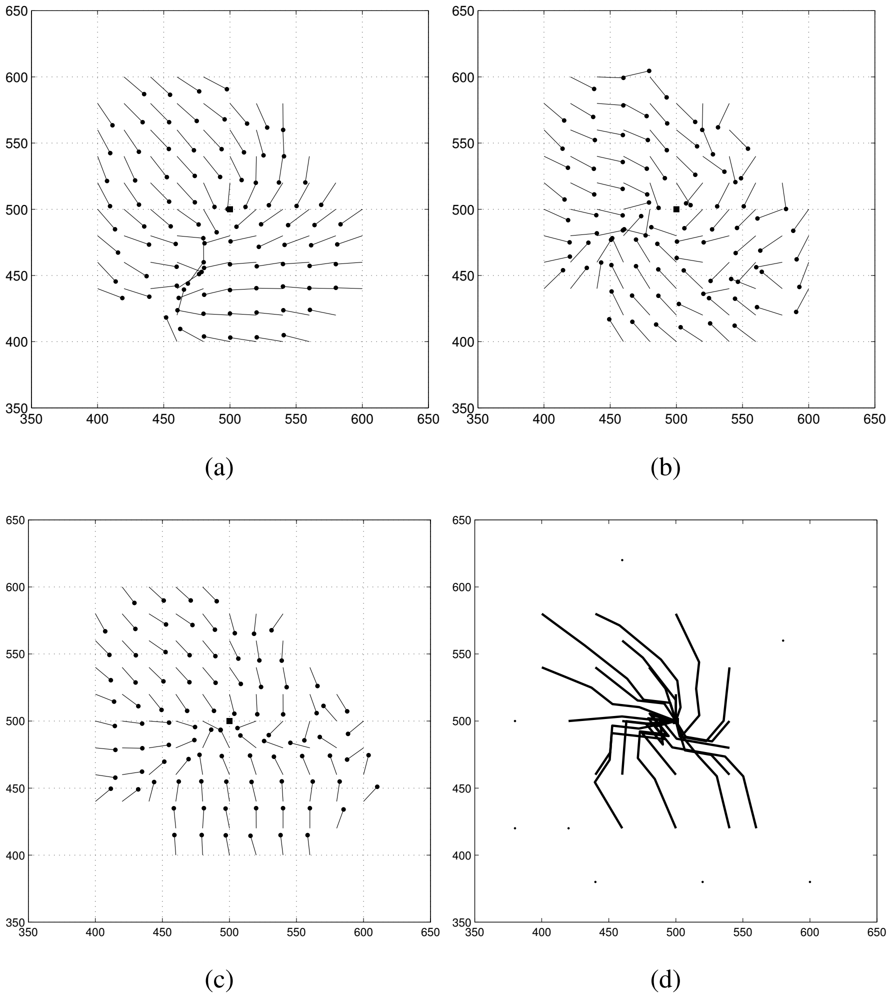
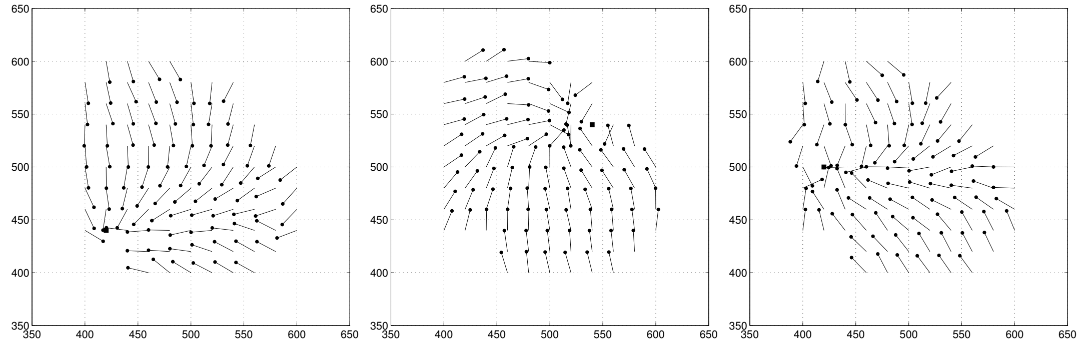


Figure 5.27: Vector maps of three different methods and the homing route of the final one. (a) is the case without PLV selection, (b) using gray information with PLV selection, (c) is using RGB information with PLV selection, and (d) is the homing route plot of (c).

We performed a robot experiment for method 2. Figure 5.27 shows the first output. There are three vector maps (a to c). The output (a) obtained using all gray information without PLV selection shows some homing tendency, but the agent does not reach home. (b) is the method using gray information with landmark selection. We manually selected the 200 pixels with high values. Then there is no spinning pattern in the homing routes. (c) is based on the RGB information with the sorting method, and it shows stability and accuracy in homing navigation. (d) shows the homing courses based on output (c). The agent uses m_3 in all the figures. We performed some additional experiments by changing the home position. Figure 5.28 shows the various outputs, and we can check the overall homing convergences.



(a) (b) (c)

Figure 5.28: Vector map using RGB information and sorting method with changing home positions. (a) 42,44 (b) 54,54 (c) 42,50 , square means home position

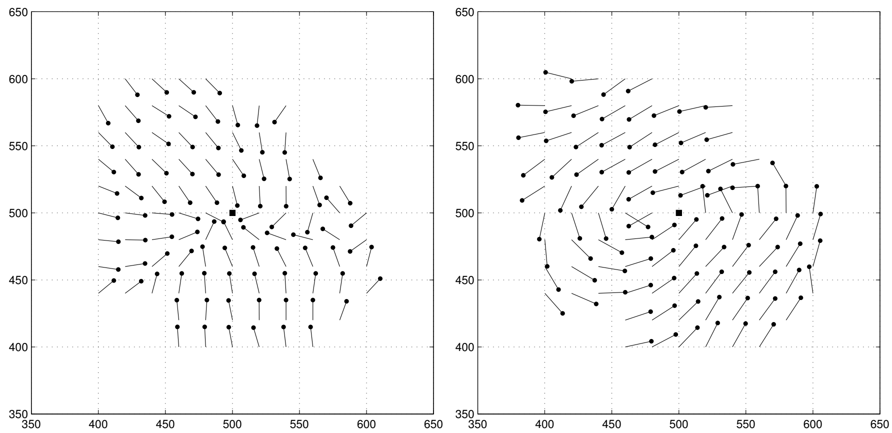
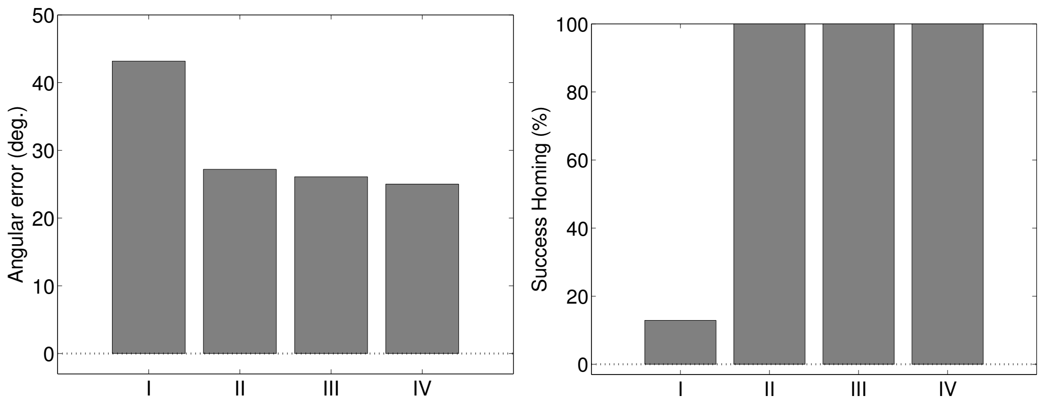


Figure 5.29: Different output according to the selected model. The left one uses m_3, and the right one shows the output obtained using m_1.

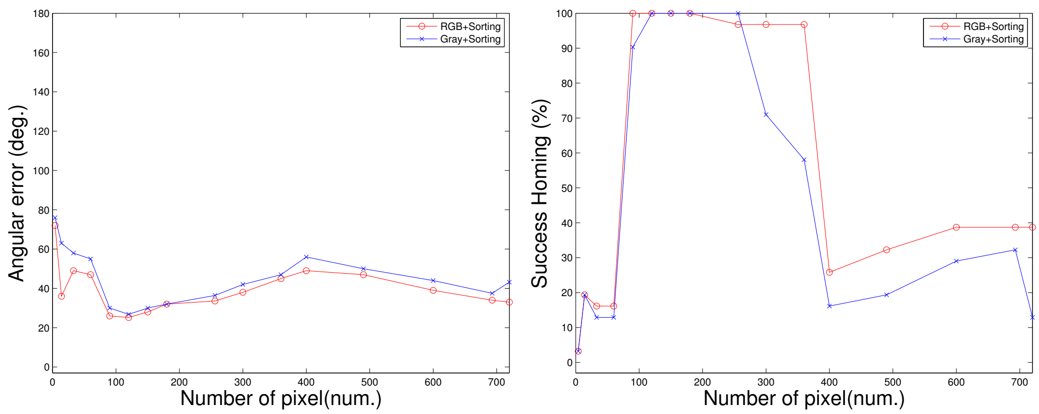
Next figure shows the effect of the selected model type. Figure 5.29 shows the output of two models. The left one is equal to (c) of Figure 5.27. One is m_1, and the other is m_3. These outputs are similar to the simulation output.

To compare the performances of these methods, we present performance graphs. Figure 5.30 shows that the case with RGB with sorting has the highest score for both homing and accuracy. The case with gray without PLV selection has the worst performance. However, the other methods such as using a simple threshold value or sorting method have successful homing ability. In this course, the selection of the number of pixel-wise landmark is very important and critically influences the output. Then, in the next figure, we show the effect of the number of pixels. In Figure 5.31, we show the effect for both RGB and gray with PLV selection method. First, the angular error of



(a) (b)

Figure 5.30: (a)Angle error performance, and (b)homing rate performance with different method. I is the case using gray information without PLV selection; II is gray with simple color threshold; III is gray with selection; and IV is RGB with selection



(a) (b)

Figure 5.31: The homing rate and angular error performances obtained by changing the number of pixels. (a) Angular error performances of both gray and RGB by the sorting method, and (b) homing rate performance of both by the sorting method

the homing vector are similar in both cases, but RGB with the sorting method performs slightly better than the gray with sorting. The lowest point shows the highest accuracy, but generally, the overall accuracy is good for all cases. The homing rates are also compared. We checked the effect of using RGB information and show the broad domain of the successful homing, which is broader than that of the gray one.

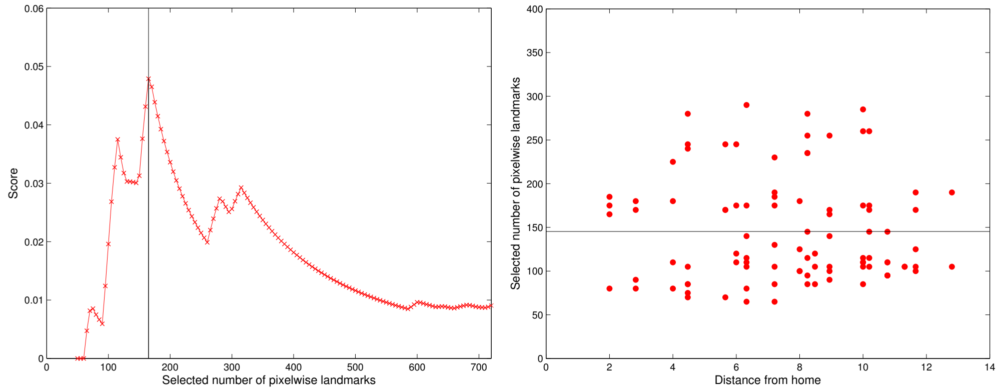


Figure 5.32: Effect of automatic pixel-wise landmark selection. The left figure shows the example of calculation of the score of PLV selection between (54,58) and (50,50). The black line indicates the selected number with the maximum score (179). The right one is the experiment on the effect of distance between two images in PLV selection. The mean of the calculated optimal output is 146

5.2.1.1 Automatic pixel-wise landmark selection

This experiment deals with the performance of the automatic pixel-wise landmark selection algorithm. In the previous section, a graph on the performance according to the number of PLV was presented. All the error and homing rate values were calculated via manual experiments. Then the output of our automatic algorithm should be in the proper region in both performance graphs. Figure 5.32 shows the output. The left graph shows one example of calculating the optimal number. Our algorithm gathers the scores for all test pixel numbers (x-axis) and the counted score (y-axis). The final selected number is marked in black. In addition, we plotted multiple cases in the right figure. The x-axis shows the distance between home and the location of the image, and the y-axis shows the calculated optimal number of PLVs. Finally, the mean of these outputs is 146, and this value is in the safe region in Figure 5.31.

5.2.1.2 Comparison with other model

we compare our algorithms with the DELV model. Figure 5.33 is the output with two graphs. These data are generated from robot experiment data. There are five methods, and the first graph shows that the best one is full-DELV. However, model 2 with RGB information yields almost identical performance. The second one shows the calculation time for each method. However, our model has extremely small operation time, except

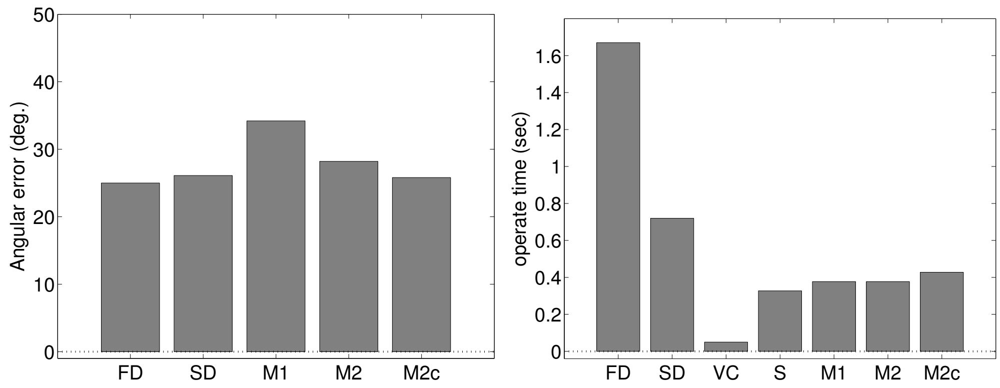


Figure 5.33: Performance comparison between our models and DELV. FD is full DELV using large computation; SD corresponds to DELV. M1 is method 1; M2, method 2; and M2c, method 2 using RGB information. VC is the visual compass, and S is the selection algorithm. The left graph corresponds to angular errors, and the right graph to operation time.

for the selection algorithm. The values for M1, M2 and M2c include both VC and S values. Thus, we see that our model is simpler and less accurate than DELV.

5.2.2 Experiments Using DTW Matching

5.2.2.1 The effect of finding stationary points via full search of DTW

Through the above parts, we suggest new visual homing techniques using both optical flow information and DTW based on snapshot model. There are two variations in calculation method about the stationary feature location. The one uses only pixels around horizontal line (1D) and the other uses entire pixels (2D) of the images. Next, there are two variations in calculation of homing direction by using DTW. The one divides image into two parts called wing matching and the other uses entire image called unifying matching. We make three variations by combining those cases. First one (I) uses both 1D information in finding stationary points and wing matching in DTW. Second one (II) uses both 1D information in finding stationary points and unifying matching in DTW. Third one (III) uses both 2D information in finding stationary points and unifying matching in DTW.

These new models are compared with one of state-of-art techniques. These are the DID (IV : Descent in Image Distance) method (Zeil et al., 2003; Möller and Vardy,

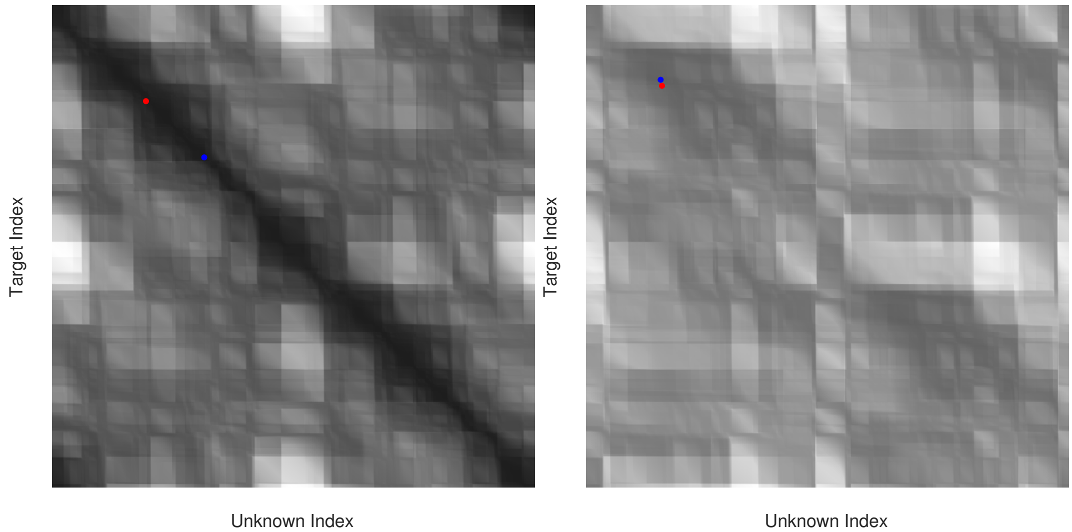


Figure 5.34: Example of the performance of our model compared with the one of full DTW search. Each colored square shows the full search output. Y-axis is the index of target angular location of the home snapshot and X-axis is the index of unknown angular location of the current view. Colored dots show the output distance of DTW where the darker one represents a more accurate method than the other. The left figure is for the case between snapshot (5,8) and (5,9). The second figure is for the case between snapshot (5,8) and (0,16). The red dot represents the full search and the blue is for our model based on finding stationary points

2006), MinWarping algorithm (V) (Horst and Möller, 2017; Fleer and Möller, 2017), COMALV (VI : Center-Of-Mass ALV) (Mangan and Webb, 2009; Hafner, 2001) and MCOMALV (VII).

Before starting our model, we check the performance of the method for finding stationary points. Our assumption is that this part can replace the full search of DTW, so we compare the output of our method with the full search DTW, which requires extensive calculation.

Figure 5.34 shows the comparison output. There are two figures. The left one is for similar inputs which are located in close proximity. Overall pattern shows darker which means the scores of DTW are generally small and the output has some differences. The right figure indicates results using inputs located at a distant location. The pattern is brighter than in the left figure, which means the errors of DTW are commonly high while the calculated location based on indexes has small differences, indicating that our model is capable of reasonable performance, even for distant points. Although

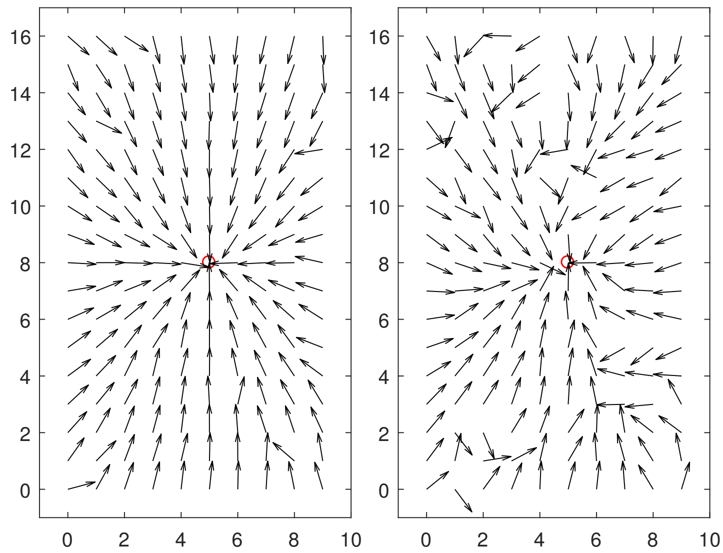


Figure 5.35: Experiment about wing matching model (I). The left figure is the pre-aligned case and right is aligned by applying the visual compass algorithm. Red circle indicates the home position and each arrow represents the homing vector with unit length at each location.

there was error in the position estimation of the left figure, the difference was small in the first place. The overall computation of full search DTW took almost 3 hours, which indicates we cannot use it directly in our model. Therefore, our model based on finding stationary points demonstrates robustness when compared with full search and we use it with alignment to find the start point of DTW.

5.2.2.2 Modified Dynamic Time Warping method based on the Wing matching method

We test our first model (I) using both 1D stationary point selection and the wing matching algorithm. In this part, we use the calculated angular locations of both stationary points and apply them into DTW matching of two wings. Figure 5.35 shows the outputs. The left figure with pre-aligned inputs shows almost perfect homing navigation. However, the right one with non-aligned inputs shows poor homing performance because of alignment errors. But all the points in that figure show successful homing navigation in the end except for (1,0). Those routes have a bit of error and cannot go straight, but they eventually reach home with a little detour.

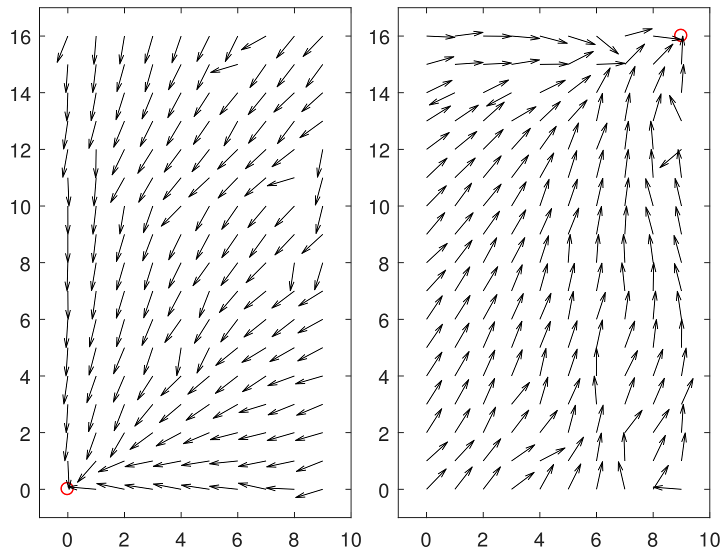


Figure 5.36: Experiment of changing home position. Home position is changed with pre-aligned inputs. The left figure uses (0,0) as the home position and right uses (9,16) as the home position.

Attached to them, we test our algorithm with different home positions. Figure 5.36 shows the outputs. There are two different cases with pre-aligned inputs. Note that our model is very robust to pre-aligned inputs with various home positions. Then, we try to check the effect of each part in our algorithm.

5.2.2.3 Modified Dynamic Time Warping method based on the unifying matching method

We observe that there are some difficulties in non-aligned inputs. Next, we test our second method (*II*) using both 1D stationary point selection and unifying DTW. It uses a four times larger part in DTW matching than wing matching to obtain higher accuracy. Figure 5.37 shows the outputs. The two vector maps generally produce better outputs the than wing matching method in accuracy. The unifying method, which takes more than twice the computation amount than wing matching, is expected to produce in better results.

Next, we test our third method (*III*) using both 2D stationary point selection and unifying DTW. It requires the most large computations than others. Because it tries to find stationary points by expanding search areas from 1D to 2D of input images. We expect

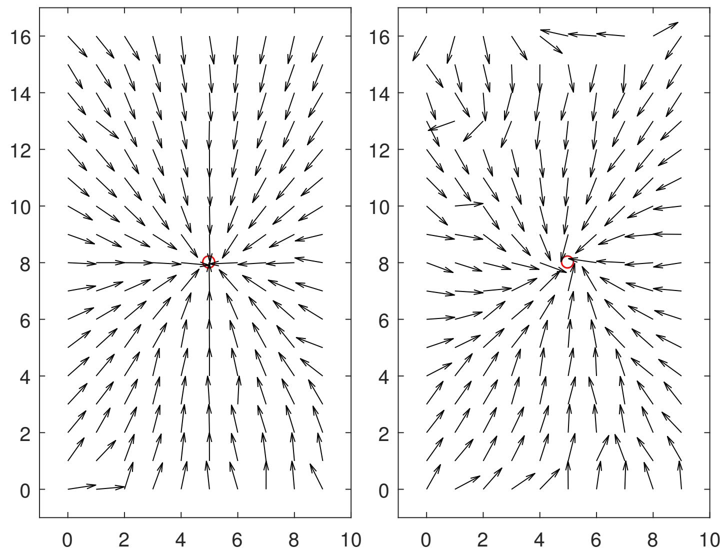


Figure 5.37: Examples using the unifying model (*II*). The left figure uses pre-aligned inputs and right one uses non-aligned inputs.

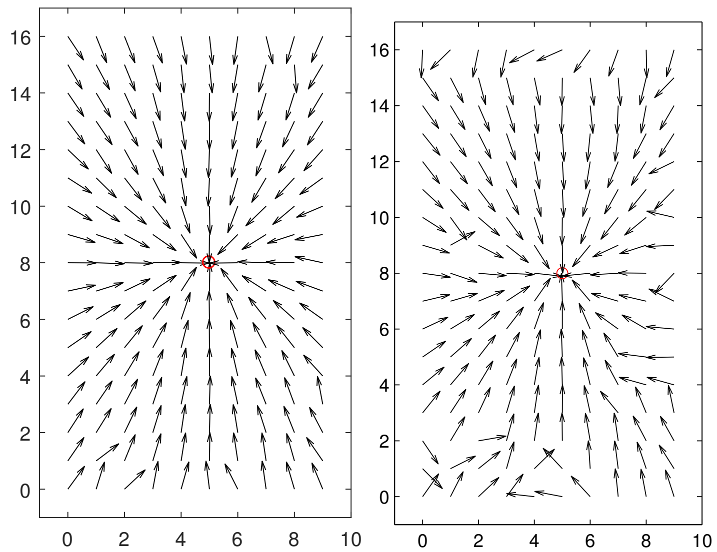


Figure 5.38: Examples using the optical flow matching model (*III*). The left figure uses pre-aligned inputs and right one uses non-aligned inputs.

that this model have the most accurate outputs. Figure 5.38 shows the outputs. The two vector maps generally produce the best outputs than our other methods in accuracy.

However, we cannot easily say that the unifying model is always more robust than the wing matching model. Figure 5.39 can serve as an example. It shows experiments comparing the three methods in different home locations with pre-aligned inputs. In this case, the home is located in the corner, so it is necessary to consider it from far away. Different to above outputs, the wing matching method shows the best record in

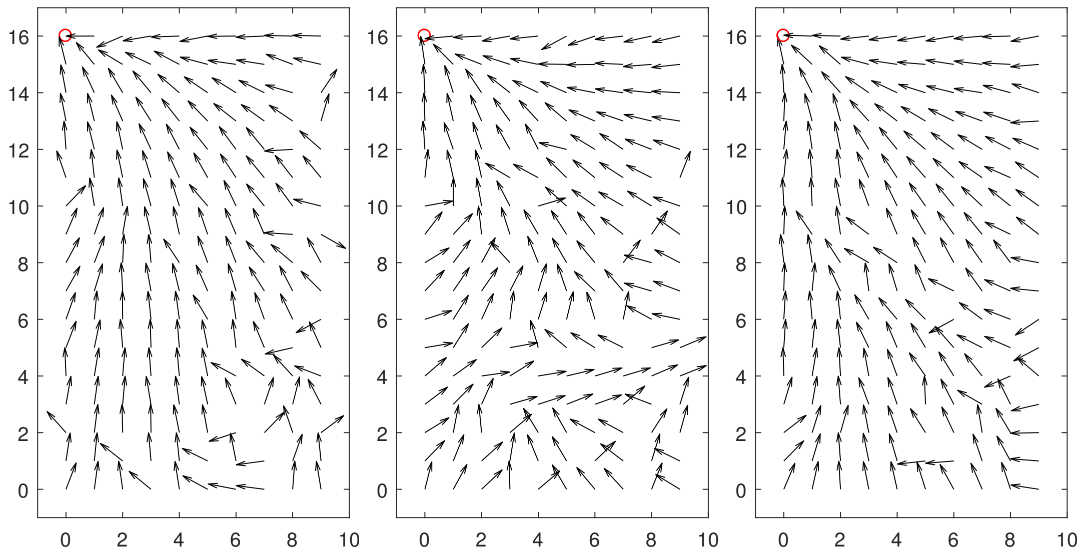


Figure 5.39: Examples comparing wing matching (I), unifying matching (II), optical flow matching (III) for a distant home location. The left figure shows the wing matching algorithm with (0,16) as the home position, middle one for unifying matching and right one for unifying matching based on optical flow with the same home position.

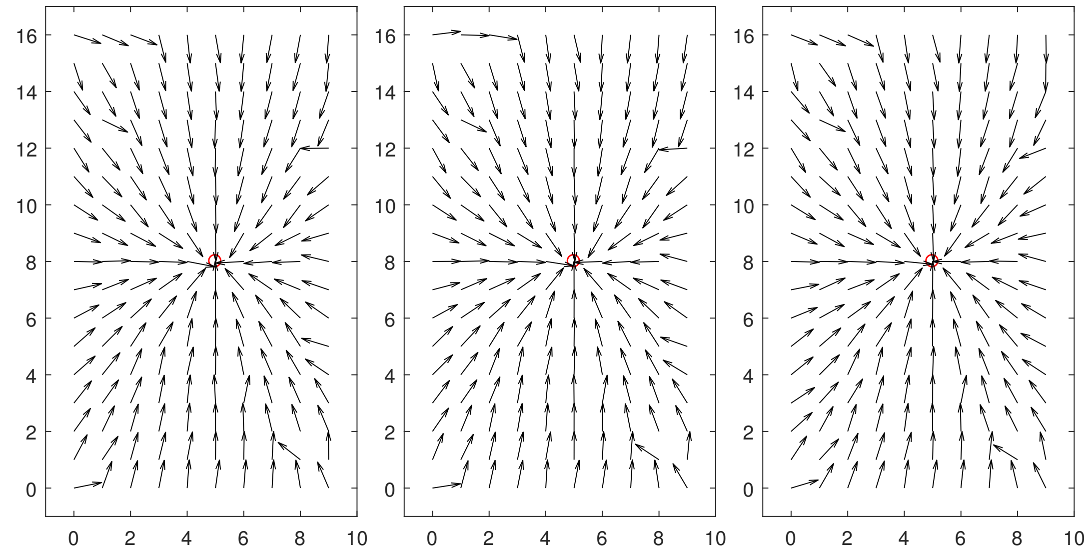
this condition. Looking at the previous results, the wing matching method seems to be more effective in returning home from a further distance although some errors cause detours from the optimal path.

Effect of the inhibition into connected edges of DTW

In this section, we try to check the effect of connected edge inhibition. Figure 5.40 shows the outputs. Based on the output with one inhibition (the left one of Figure 5.35), the other methods using non-, reverse- and both inhibition show poor performance. Therefore we find that using a reasonable inhibition such as the method shown in Figure 5.35 is helpful in navigation.

Effect of the searching area

In this section, we test the effect of the size of the searching area. With non-aligned inputs, we have to use the visual compass to align them with the snapshot image and it can cause some errors in alignment. We increase the search by additional small rotations of input images to find optimal pairs in order to reduce the alignment error. Figure 5.41 shows the output with various searching area sizes. What is interesting here is that seeing unconditionally large parts of an area does not produce better results.



(a) (b) (c)

Figure 5.40: Experiment of the effect of inhibition to connected-edges. the left figure is without inhibition and the middle one is reverse inhibition. The right figure uses both inhibition and reverse inhibition. The left side of Figure 5.35 is the case using one inhibition with the highest accuracy.

5.2.2.4 Comparison with other algorithms

Figure 5.42 shows vector maps to compare our approaches with other four benchmark algorithms. Interestingly, in pre-aligned condition, our third one (*III*) has the best record in homing accuracy than the other approaches.

Table 1 shows homing performance for our three models, DID, MinWarping, COMALV and modified COMALV method in the 'A1original' environments.

5.2.2.5 Applying into different environment

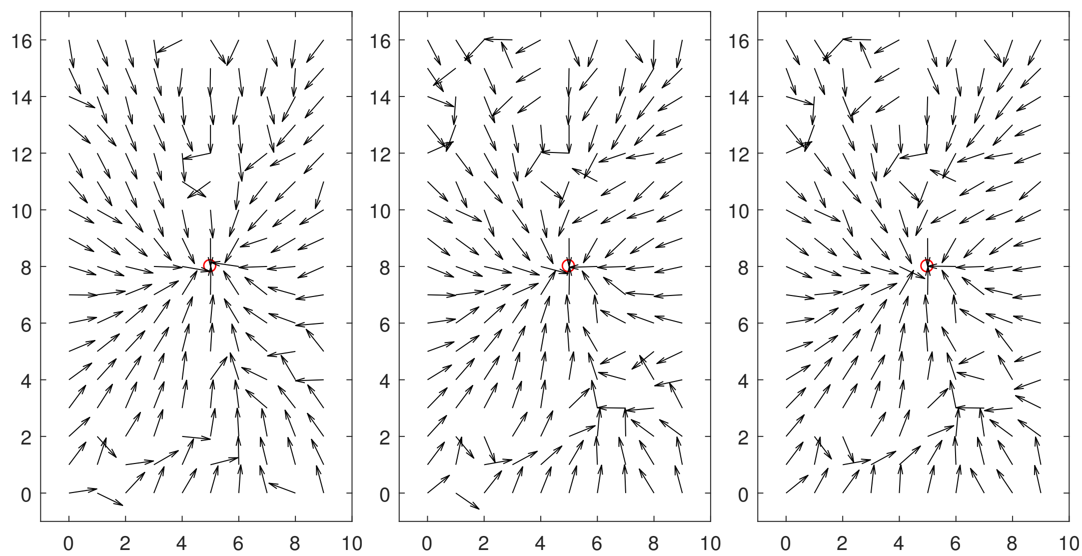
Also, we test the adaptability of our algorithm by using different image databases. As mentioned in the environment discussion, we use 110 additional pictures from our lab environment, and we apply our model to this data. Figure 5.43 shows the output for both aligned inputs and non-aligned inputs. We verify successful homing navigation through the results and confirm that our model is adaptable to other environments.

Table 5.1: angular errors of different methods in 'a1original' environment

		error μ ($\pm\sigma$)	N	$0 \leq \varepsilon_\theta < 45^\circ$	$45 \leq \varepsilon_\theta < 90^\circ$	$90 \leq \varepsilon_\theta < 180^\circ$
With reference	*I*	6.33 ($\pm$0.97)	169	99%	1%	0%
	II	5.71 ($\pm$0.92)	169	99%	1%	0%
	III	4.89 ($\pm$0.75)	169	100%	0%	0%
	IV	23 ($\pm$2.6)	169	92%	7%	1%
	V	11 ($\pm$1.5)	169	99%	1%	0%
	VI	19 ($\pm$2.0)	169	98%	2%	0%
	VII	25 ($\pm$3.3)	169	86%	13%	1%
Without reference	*I*	13.15 ($\pm$3.0)	169	93%	6%	1%
	II	14.2 ($\pm$2.7)	169	95%	3%	2%
	III	12.8 ($\pm$2.8)	169	95%	4%	1%
	IV	29 ($\pm$3.4)	169	83.4%	14.2%	2.4%
	V	11 ($\pm$2.1)	169	99.4%	0.6%	0%
	VI	21.7 ($\pm$3.5)	169	94.1%	5.9%	0%
	VII	38.2 ($\pm$4.9)	169	70.4%	18.9%	10.7%

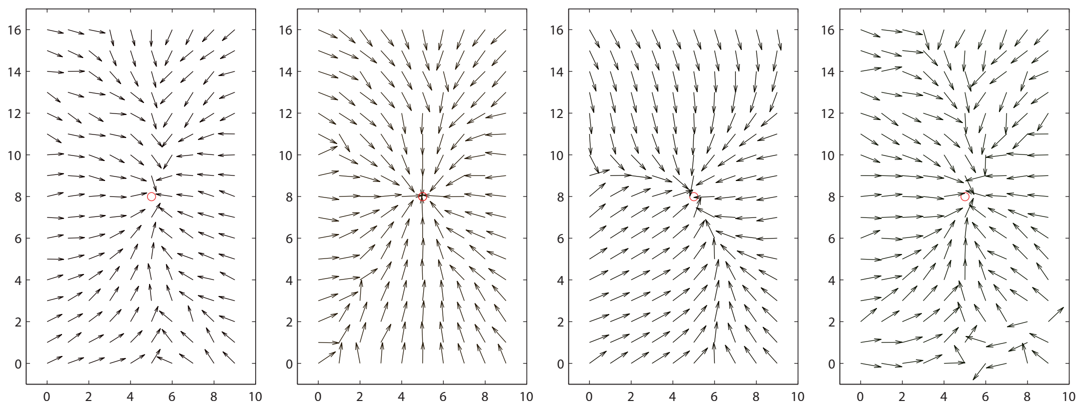
Table 5.2: angular errors of different methods in our lab environment

		error μ ($\pm\sigma$)	N	$0 \leq \varepsilon_\theta < 45^\circ$	$45 \leq \varepsilon_\theta < 90^\circ$	$90 \leq \varepsilon_\theta < 180^\circ$
With reference	*I*	12.33 ($\pm$1.46)	125	98%	2%	0%
	II	11.7($\pm$1.7)	125	97%	3%	0%
	III	10.1 ($\pm$1.9)	125	100%	0%	0%
	IV	25 ($\pm$3)	125	94%	6%	0%
	V	61 ($\pm$9.5)	125	45%	25%	26%
	VI	49 ($\pm$7.4)	125	56%	29%	15%
	VII	35 ($\pm$5.4)	125	72%	25%	3%
Without reference	*I*	15.9 ($\pm$3.5)	125	92%	6%	2%
	II	16.9 ($\pm$3.1)	125	93%	6%	1%
	III	16.2 ($\pm$3.4)	125	91%	7%	2%
	IV	23 ($\pm$5.2)	125	94%	6%	0%
	V	60.22 ($\pm$9.7)	125	48%	26%	26%
	VI	58.25 ($\pm$7.9)	125	47%	34%	19%
	VII	32.9 ($\pm$6.5)	125	81%	13%	6%



(a) (b) (c)

Figure 5.41: Experiment of the effect about searching area size. The left side uses a small searching area ($\pm 0 deg$) and middle one uses an intermediate area ($\pm$ 15 deg with 3 samples). The right figure uses a wide area ($\pm$ 15 deg with 6 samples). The one in Figure 5.35 uses a wider one ($\pm$ 15 deg with 10 samples)



(a) (b) (c) (d)

Figure 5.42: Comparison of DID (IV), MinWarping (V), COMALV (VI) and MCOMALV (VII) method in vector map in ORIG environment (a) the DID method (b) the MinWarping method (c) the COMALV method (d) the modified COMALV method

Table 1 shows homing performance for our three models, DID, MinWarping, COMALV and modified COMALV method in the 'our lab' environments.

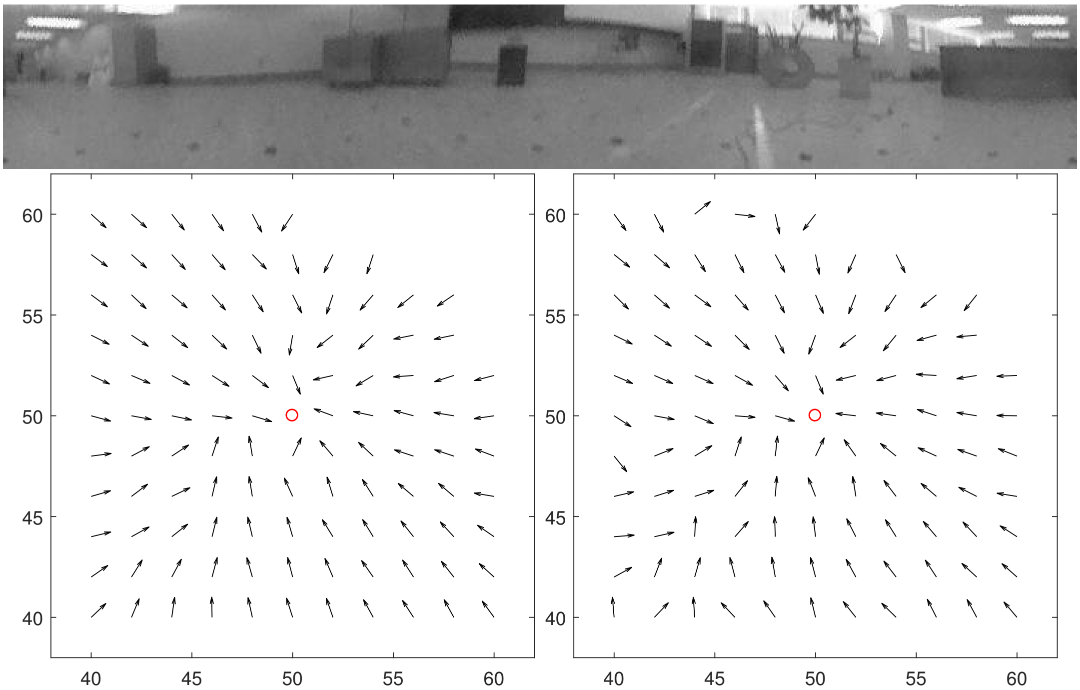


Figure 5.43: Experiment using different environments. Above is the example panoramic image of our environment which is used as a snapshot. The lower left figure is the output using wing matching with pre-aligned inputs. The lower right figure uses the wing matching with non-aligned inputs. Red circles in both outputs represent the home position.

5.2.3 Experiment with Illumination

We know that our vision based models can be affected by the illumination in the environment. Therefore, we put perform experiments during four different times of the day. Figure 5.44 shows four cases. The first case was studied at 6:00 AM with limited sunlight. The second case was studied at 2:00 PM with both sunlight and light from the room. The third case was studied at 8:00 PM with light from the next room only. The fourth case was studied at midnight with light from both rooms. In this part, we evaluate the performance difference between sunlight and the light from fluorescent light bulbs on the ceiling. These factors affect the brightness in the measured visual cues.

The output is shown in Figure 5.45. There are two vector maps that show the calculated homing directions of each of the two methods. The left map shows our model using DTW and has a smaller error than the right map which uses MinWarping. These maps were produced for the study at 6 AM. Another map showing the performance changes

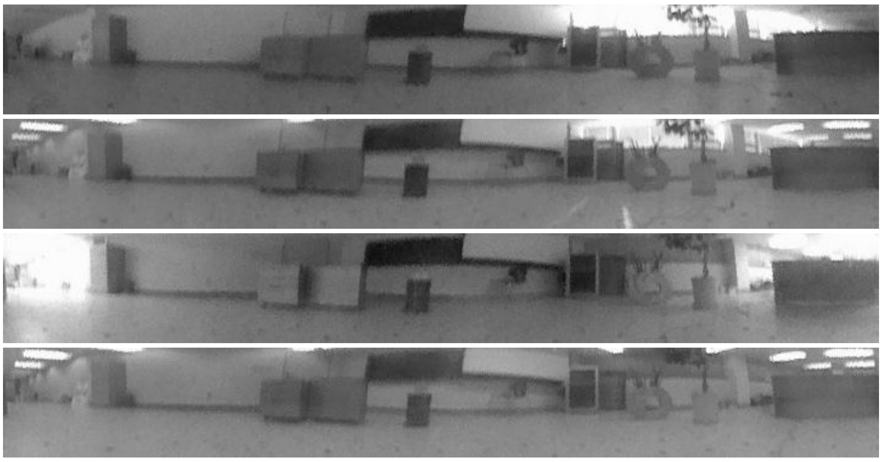


Figure 5.44: Images of the experiment at different times of the day. The first image was taken at 6:00 AM, the second image was taken at 2:00 PM, the third image was taken at 8:00 PM, and the fourth image was taken at 12:00 AM

over time shows that our model produces a better output during all times of the day in our environment. Thus, our models above have partial robustness in dynamic, occluded, and illuminated environments without the use of optimization techniques.

5.2.4 Experiment with SLAM-like Loop Closing

Additional, we apply our snapshot-based model into the wide region. We chose the loop closing problem which is widely mentioned in SLAM techniques. Loop closure is a problem in recognizing a previously visited location and updating the accumulated measurements. This can be crucial in decreasing errors by updating accumulated errors in dead reckoning. Based on odometry, dead reckoning (path integration) is the process of calculating the current position by using a previous position and advancing that position based on known or estimated speeds. In this section, we use both distance(d) and rotation angle(θ). Then it can be simply calculated by

$$\theta_t = \theta_{t-1} + d\theta_t$$

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \end{bmatrix} + d_t \begin{bmatrix} \cos(\theta_t) \\ \sin(\theta_t) \end{bmatrix}$$

When a match is detected, it uses an algorithm to compute the similarities in sensor measurements and reset the current location based on the information from the previ-

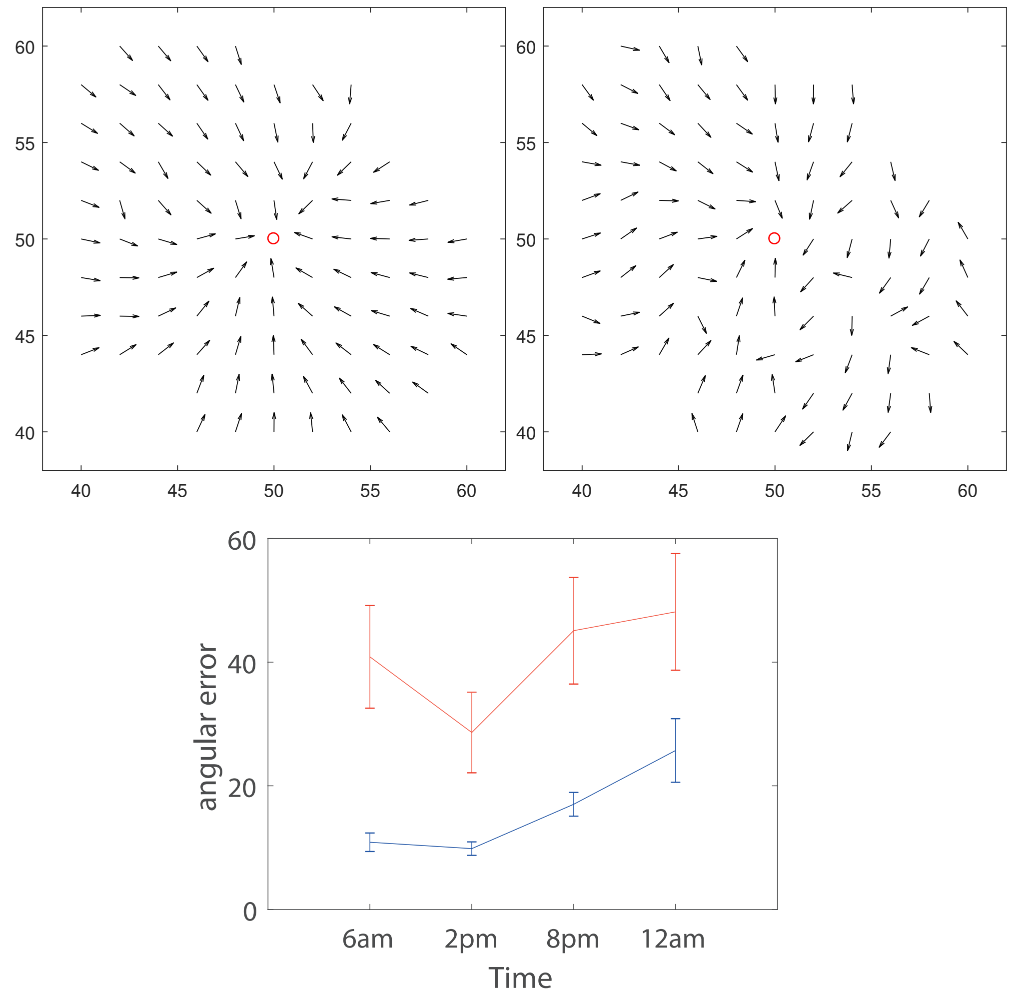
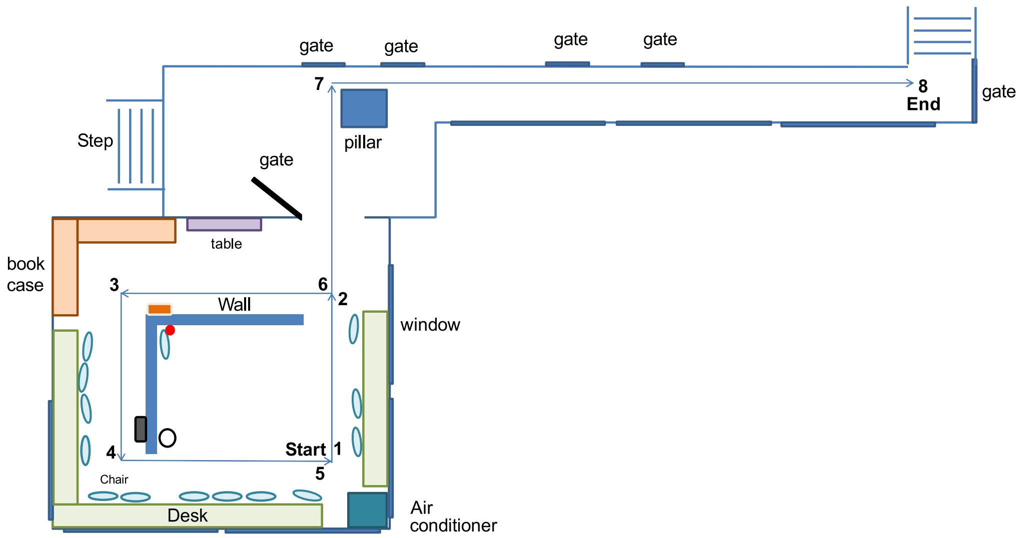


Figure 5.45: Output example of all day experiments. The left image shows the use of our model based on DTW and the right image shows the use of MinWarping. The red circle at (50,50) shows the home location and the black arrows show the calculated homing directions. The lower image shows the changes of the overall homing performances over time. The blue one is about our method using DTW and the red one about MinWarping.

ously visited location. For example, this information can be extracted by comparing features in the memory of the robot from each previously visited location.

Then, we apply our image difference concepts in comparing features instead of using complex visual features. There are three parts in our algorithm. The first part is the measurement of visual cues. The algorithm attempts to find the difference between two neighboring images and match them to the current image to check the necessity of additional snapshot gathering. If the accumulation of differences is larger than threshold, the algorithm assumes that the robot has left the local area including past snapshot



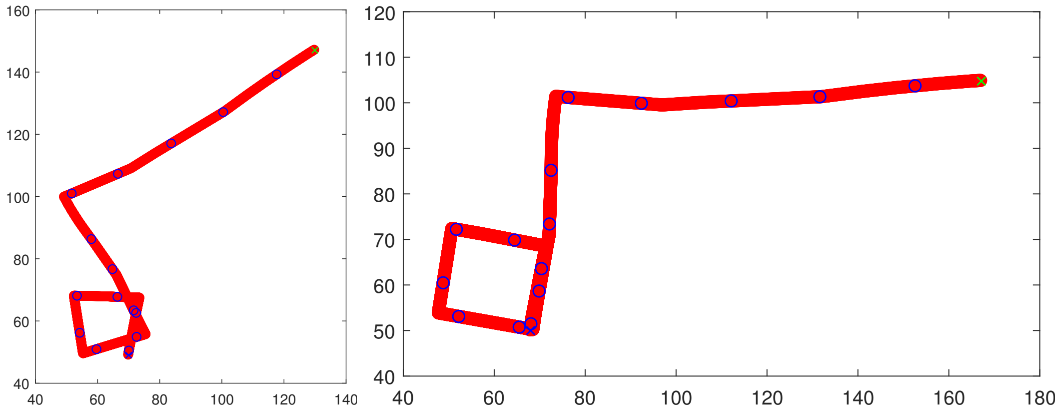


Figure 5.46: Loop closing test in the room and in the corridor environment. The image on top is the illustrated top view. It shows the object location in the room and includes the path of the robot moving. The following figures show the effect of loop closing using simple image differences used in our models. Marking numbers ($1 \sim 8$) show the corner locations in robot route. First figure is the path without loop closing (using only odometry information). Second figure shows the use of loop closing. The blue circles in the lower images show the locations of new image acquisition.

and record another snapshot. The second part involves movement. The algorithm attempts to update the location information and records the path. The third part is the overall matching to past ones when loop closing Occurs. Here, we decide that moment by calculating differences between older images ($diff$) and the recorded snapshot. It can by calculated by

$$Diff = \min_{\theta} \sum |I_0 - rot(I, \theta)|$$

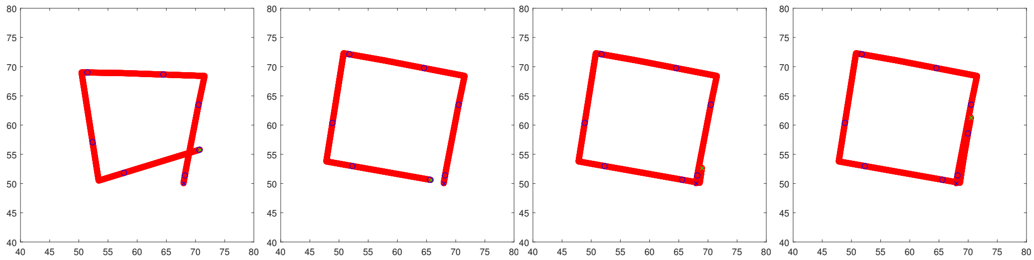


Figure 5.47: Examples of changing routes around loop closing. First one is before loop closing and second one is right after loop closure. Third one and fourth one are next examples after route update.

where I_0 is the one of the recorded snapshot and $rot(I,\theta)$ means rotating current image. If after traveling some distance, the image difference between the targeted past snapshot and the current image shows a small difference (under 50 percent of threshold used in additional snapshot gathering), it tries to update the past route by updating its current location and renewing the odometry.

To build loop closure, we make a simple assumption. The assumption is that there are two independent errors in both rotation and distance with cumulative characteristic. First, we perform alignments based on the rotation difference from the visual compass. Then we can find the estimated angle (θ_T) based on measurement . We can then find the rotational error compared with the cumulative angle.

$$
\begin{aligned}
d\theta_T &= \arg\min_{\theta} \sum |I_0 - rot(I,\theta)| \\
\theta_T &= \theta_0 + d\theta_T = \theta_0 + \sum_{i=1}^{T} d\theta_i K_\theta \\
K_\theta &= \frac{d\theta_T}{\sum_{i=1}^{T} d\theta_i}
\end{aligned}
$$

In addition, as our rotation error modeling is designed as constants to $d\theta$, the rotation error component (K_θ) can be calculated. Next, after alignment update, we find the distance error component (K_D) making estimation of location more accurate. Then we can rebuild the all path.

$$
\begin{bmatrix} \overline{x_t} \\ \overline{y_t} \end{bmatrix} = \begin{bmatrix} \overline{x_{t-1}} + d_t K_D \cos(\theta_0 + \sum_{i=1}^{t} d\theta_i K_\theta) \\ \overline{y_{t-1}} + d_t K_D \sin(\theta_0 + \sum_{i=1}^{t} d\theta_i K_\theta) \end{bmatrix}
$$

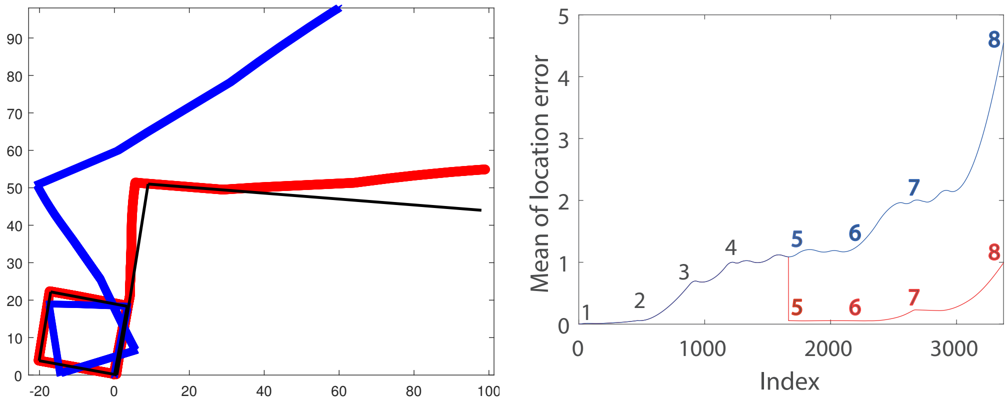


Figure 5.48: Performance evaluation. Three routes are shown on the left. The blue one is the output of dead reckoning, the red one with loop closing and the black one is the desired output. The right one shows the location error. Marking numbers (1 8) show the locations of robot in the Figure 5.46. The “Index” is the frame number and the large drop at around 1700 (marked with ”5”) is the moment of loop closing.

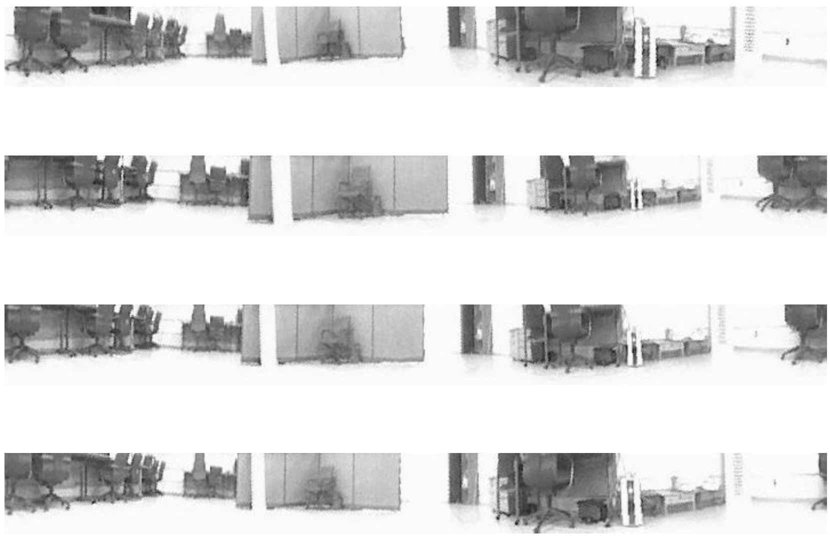


Figure 5.49: Examples of panoramic images. First one is from home, others from right before loop closing in time-ordered sequence. Fourth one is the most similar one with home image at loop closing.

To test this problem, we use two environments. The first environment is a room with a corridor. There are many objects including a book case, a table, a window, an air conditioner, trashcans and so on. The partition in the center of the room acts like a wall dividing the space like a donut. Figure 5.46 shows them including robot route. During the first rotation, we open the gate and extend the field and robot goes out. The output

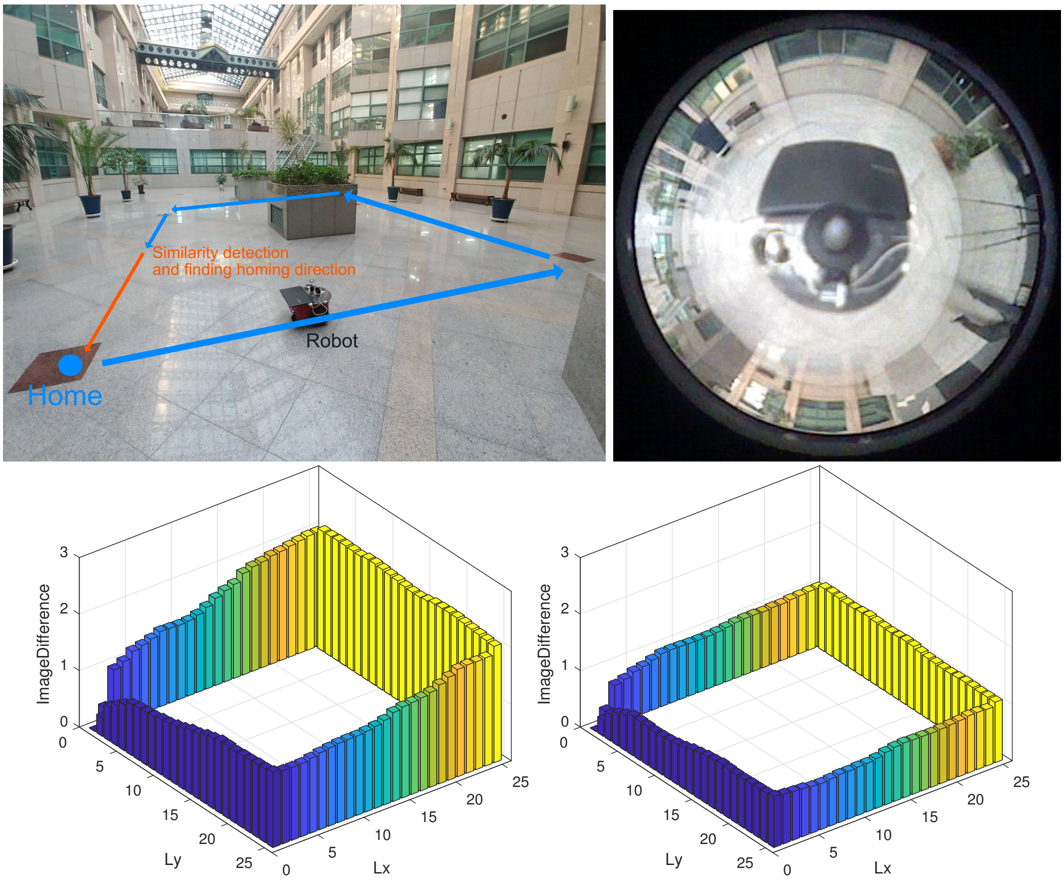


Figure 5.50: Loop closing test in hall environment. The first image shows this mission of this part. The blue one is the robot moving along the designated route based on odometry and orange one is the homing courses after finding similarities. The second image is the recorded home image. The third image is the calculated image differences according to this course and the fourth image shows the use of histogram equalization in the image processing.

without loop closure shows failure in localization by accumulative errors in odometry. However, another output with loop closure shows a successful update but fails in the corridor after update because there is no more loop closing. The blue circles show the location of an additional snapshot based on the calculated image differences.

Based on our loop closing method, detail changes around the moment of loop closure are shown in Figure 5.47. There are four cases in time-ordered sequence. First case shows a large error before loop closure. The second case shows successful alignment based on image similarity after loop closure. The third and fourth cases shows good outputs with continuous updates from past measurements. We calculate the performance in Figure 5.48. In the figure on the left, the blue one is normal dead reckoning

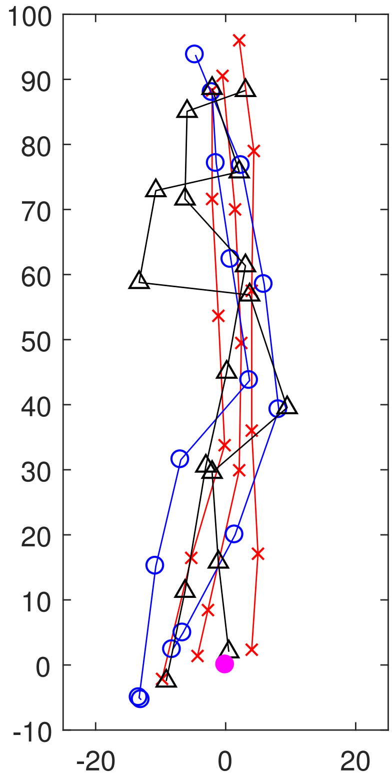


Figure 5.51: Homing sequences of using various our models. The red line shows the output of our model using DTW, the blue line shows the output for the moment model using distance estimation and the black line shows the output for the temporal update of the pixelwise model. The magenta dot at (0,0) is the home location.

and the red one uses loop closing. The black line shows the desired course and we check the errors between them. The second figure shows the output. Compared to the normal one, our model shows performance improvements after loop closure (around index 1700). Next we have focus on the visual cues between index 1600 and 1700 before loop closure. Figure 5.49 shows the aligned panoramic images. Compared with the first image from home position, second one (index 1600) and third one (index 1650) shows large differences than fourth one (index 1700) at loop closing. Our model also shows better accuracy than the normal one in a time-ordered sequence. After update, it fails in corridor region because there are new places and no more loop closing.

Next is the large open area in the hall of the building. There are many objects including flowerpots with marble structures. This environment has a large glittering on the ground and it can directly affect the output. In this environment, we designate a single task to the robot. It records a single home image and rotates this area in rectangular course. Then there must be a moment having image difference be small and sequentially find homing directions from that position. Figure 5.50 shows this mission in the first image and the second image is the recorded home image. The third image shows the calculated image differences between home and other images at each point. The fourth image is the case using histogram equalization to filter out the unnecessary illuminations.

In this section, we test three of our models shown in the Figure 5.51. The first model is based on DTW, the second model uses the moment model with distance estimation and the third model uses pixelwise model of temporal update. The performance of the first method shows high accuracy. However, the other two methods show successful homing. The model using the moment model includes unstable distance information which is estimated from the image and the other model uses simple pixel color values without matching. Despite their simplicity, the overall homing patterns are successful. With this in mind, we can check that our observations can be applied to the loop closing problem which is widely used in modern SLAM techniques.

5.3 Performance Comparison in Static Environments

We suggest new visual homing techniques with various approaches. We tested them by using simulations and robot experiments and compared their performances with other benchmark algorithms. In this section, we attempt to evaluate our methods from different views. We also attempt to integrate the outputs from the previous sections and compare them with other benchmark algorithms.

Therefore, we select representative models from each section. Through section 3, we try to apply the moment function into the homing problem to effectively combine range data with color information and check if the new moment potential has convexity. Then, we select two models using these properties. One (I) uses the moment function ($n = 2$) including both range and color of each angular direction. The other one (II) uses an area integrated version of the moment function ($n = 4$), including both range and color of each angular direction. We check for better performance than the first one by higher dimension from integration acting like clustering.

In section 4, based on bipolar approaches, we know that the probabilities of applying visual masks in the homing problem as receptor and we can apply one more characteristic of the insect "ommatidia" into the mask generation process which can largely

Table 5.3: Performances in two test environments. Angular errors with various methods including ours in two test environments (ε_θ is the absolute angular error and n is the number of test points; 1000 masks used,μ is the mean of the angular error and σ is the length of confidence interval in 95 percent. *p¡0.05).

		I	II	III	IV	V	VI	VII	VIII	IX	X
ORIG	μ	14	11	16	9	6	5	23	11	19	25
	$\pm\sigma$	3.1	7.4	1.8	1.2	1.0	0.8	2.6	1.5	2	3.3
	$0 \leq \varepsilon_\theta < 45°$	100	100	100	99	99	100	92	99	98	86
	$45 \leq \varepsilon_\theta < 90°$	0	0	0	1	1	0	7	1	2	13
	$90 \leq \varepsilon_\theta < 180°$	0	0	0	0	0	0	1	0	0	1
OURL	μ	6	8	25	20	12	10	25	61	49	35
	$\pm\sigma$	1.4	3.6	4.4	2.6	1.2	1.9	3	9.5	7.4	5.4
	$0 \leq \varepsilon_\theta < 45°$	94	45	56	72	100	100	77	88	98	100
	$45 \leq \varepsilon_\theta < 90°$	6	25	29	25	0	0	22	12	2	0
	$90 \leq \varepsilon_\theta < 180°$	0	26	15	3	0	0	1	0	0	0

enhance robustness. There are two selections. One selection (III) uses only one reference image while the other selection (IV) uses three reference images like the DID model to mimic simple learning around its home position.

In this chapter, we tried to interpret the basic principle of landmarks in navigation. Based on pixelwise approaches, we can find that the simple homing using pixels as a landmark can show success without object extraction based on RunDown update. Then we improve it by attaching DTW matching and we select two of them. One model (V) uses both 1D information in finding stationary points and wing matching in DTW. The other (VI) uses both 2D information in finding stationary points and unifying matching in DTW.

Compared with our models, we select four state-of-art techniques that are introduced in background. These are the DID method (VII), MinWarping algorithm ($VIII$), COMALV (IX) and MCOMALV (X). Then, we use two environments to test them simultaneously. The one is 'ORIG' from Vardy's open dataset and the other is 'OURL' that we made.

The table 5.3 shows the angular errors and the other table 5.4 shows the calculation complexity. Through the outputs, we can know that the one using DTW (2D,unifying)

Table 5.4: Calculation time

Type	Algorithm	Time(sec)	Type	Algorithm	Time(sec)
I	Moment (d=2)	0.3345	VII	DID	0.0003
II	Moment (d=4)	0.5721	VIII	MinWarping	10.0158
III	Vmask (1ref)	0.00009	IX	COMALV	0.00003
IV	Vmask (3ref)	0.0007	X	MCOMALV	0.00005
V	DTW (1D,wing)	0.788	Alignment	visual compass	0.0411
VI	DTW (2D,unifying)	1.398	Alignment	rearrangement	0.0988

shows the best record in Vardy's environment. Also, all of our other methods and MinWarping show successful output. However, the calculation time of MinWarping due to complexity is too high. In this part, we can know that both methods using visual masks and DTW shows successful performances. Next, in our environment, the moment model having the higher dimension shows the best output. This is because it uses the range data from the sensor not the image. This environment has sunlight from windows and this cause large errors in the measurements, resulting in poor performances. Despite the difficulties, our models show successful home arrival with small losses in accuracy.

According to the integrated assessment of our methods, we can see that our methods have greater performances than other benchmarking algorithms. Our complex models (II,IV,VI) from three chapters show smaller ($6° = VI < IV < II = 11° = VIII(MinWarping) < others$) errors in homing direction and also shows a lower time consumption. Also, our model can make successful homing characteristics in our environment including illumination problems. Some of the characteristics of our model are shown below. Firstly, using an accurate range sensor makes the moment model largely robust and its output shows higher accuracy than our other models. Secondly, using the visual mask generated by following the structure of ant's ommatidia, the model can operate well and it can be improved by more future works. Lastly, using DTW pixel matching shows the best outputs based on the optical flow and requires the largest computation time among our models.

5.4 Summary of Chapter 5

In this chapter, we suggest a series of new visual navigation algorithms using pixel-wise landmarks. Based on insect's vision, we made all the pixels as landmark called pixel-wise landmark. We think the insects cannot afford to exact object detection, then we try to take each retina input as landmark. Our new model use this pixel-wise landmark like object and it calculates homing direction based on changes of these pixel-wise landmarks. Similar with ALV, It converts the snapshots of both home and current into pixel-wise landmark vector sets and finds the sum of changes in landmark vectors as home direction. We show the effects of using (estimated) distance model and expand it to the one using only color. Through the experiments with mathematical proof, we investigate the convergence of our model. We find that the our model without distance cannot directly find homing direction but four types of candidates including one answer having convergence. Among those candidates, it can find right answer by following each one decreasing the image distance between home and current one. In this part, we think that if the small insect can use simple pixel matching, then it can directly find homing direction without unnecessary movements. Therefore we add simple and proper matching process like ACV that is modified version of ALV. It is DTW algorithm that can match the time-series warped data to original one. Because time series concept is replaced with angular location then matched output of DTW is equal to the one of landmark vector matching. This matching is operated between home and current pixel-wise landmarks to find angular differences. It needs adjusting start-end position and we add the concept of two static points which is not affected by movements and located at the opposite direction. Combining DTW with finding static points can make robust matching in pixel-wise landmarks. It can operate like robust ACV model using pixels as landmarks themselves in holistic approach without object extraction.

Chapter 6

Conclusion

In this dissertation, new studies of holistic matching approaches to visual navigation have been presented. We have focused on both the simplicity and robustness of insect's navigation to solve the high complexity problem of conventional navigation methods. Based on the holistic snapshot model, we suggest three new homing navigation methods using visual cues without object extraction. These methods can be classified into three groups according to chapter.

6.1 Visual Navigation Using Various Moment Models

Chapter 3 is the first method we propose and uses the moment function. Instead of using mass, we use color value to simply merge range data with visual cues in navigation. We think of each pixel as a mass object and convert pixels into a potential field. Also we find the global minimum point and condition of convergence according to the changing dimension factor. Based on these ideas, we suggest a generalized version for changing dimension number and evaluate various effects. Different to the standard moment model with cylinder-shape landmark assumption, we find that the holistic model using range distribution can be improved using a particular dimension number. We verify that both range and vision can be incorporated into navigation without object detection by using a characteristic moment function.

We think that there are some limitations. First is the form of the moment function. We think other possible variations such as the sum of C_i/r_i can operate as a collision avoidance mechanism. Different to the basic model which has unique convergence, a property of this variation model is divergence from one landmark and rapid decrease to zero. Therefore we can use this property for collision avoidance. However, we do not apply this model into our current model. We also think there must be other variations of the model having effective abilities. We will research those variations as future work.

Second is modifying our new alignment using a triangulation method by using three convergence points. Different to other compass models such as visual compass, landmark arrangement, warping method, etc., the proposed model requires only three calculations. Based on unique convergence of the moment model, each convergence point has uniqueness in spite of variations in both angular and transitional position. We can find the intersection point of the candidate by comparing three radii which have circular distribution from the center, similar to a triangulation method, but in this part, there are errors in homing sequence without a compass. In these cases, measurement about three convergence points has some errors due to the approximations involved, so we will seek to upgrade our alignment model to have robustness.

Third is applying two boosting processes. One is applying feature matching to our moment model similar to SIFT to remove some occluded parts. As we do not use any data process filtering or matching of inputs, there are many needless or occluded areas of an image. However, our model does operate well and we can check convergence, We will evaluate increasing the robustness of our model by applying additional processes. The other is the unifying algorithm to multiple moment models. Because our moment function can be expanded into various forms and applied to various sensors, it should be possible to reduce the overall problems by the fusion of multiple moment measurements if sensors are added.

6.2 Visual Navigation Using Visual Masks

Chapter 4 discusses the second method using a mask receptor. Inspired from the receptor cells in the retina, we create a set of organized masks in the input image space and use both the location and score of each mask to build a landmark vector. These bipolar-like cells are generated by the combination of Haar-like features into various types. First we systemically locate simple masks with vertical edge detectors having various sizes as levels. We noted that changes in mask size can affect robustness. Therefore we add some randomness in mask generation to have various types and sizes. This random property exhibits successful homing in simple environments but not in occluded ones. The first model using this randomness calculates the matching score differences with a set of Haar-like masks and then estimates the homing vector. When each mask is expressed as a landmark vector whose length is set to the matching score, the averaged landmark vector can represent the snapshot at a given location. The difference of the two averaged landmark vectors can determine the homing direction, which is equivalent to the net effect of the matching score difference with a set of masks. In this part, we observe the linear proportionality of the sum of the mask scores to distances from home and we apply multi-reference images to find image differences.

Another model we suggest is to use the matching score differences between one of three snapshots near the home location and the snapshot at the current location. The matching score difference approximately estimates the distance between a pair of snapshot spots, and the relative distance of matching score differences can determine the homing direction. Our results show that a collection of random Haar-like masks represent the characteristics of a snapshot image well. Then our second model exhibits robust homing performance. Sophisticated feature extraction is not required for visual homing, but a set of simple masks are sufficient to find the visual features, especially to derive the homing direction. In short, we verify that holistic visual navigation can be operated with both freely distributed masks and multiple images around home, similar to the learning phase of insects.

We think that there are some limitations. The first problem is the lack of biological analysis. As our model is based on a bio-inspired bipolar cell structure, our model has to focus on the biological structure of the vision system. In prior research, it is argued that the bipolar cell is the basic unit of vision and the higher functions are combined in the visual cortex. But in our model, higher functions can also be located in retina positions before the brain. Therefore we will try to determine a robust algorithm from biomimetics to explain the overall navigation principles of various creatures.

The second problem is mask generation. Compared with receptors in retina, our model uses a small number of masks and generated masks are much larger than ones having small FOV. We need to substitute the random arrangement of generated masks by applying some rules of bio-inspired cell generation. In essence, we have to generate a well-organized artificial retina system, Similar to the compound eye of dragon fly, our model has to be modified to having a structure of well-organized masks. We think that both a larger number of masks and a systemical arrangement of same can produce improved results than the current method. With this artificial receptor array, we emulate an insect's perspective.

We have to evaluate the recognition properties of this artificial retina.

The third challenge is finding the optimal mask arrangement. This challenge involves customized mask generation for better homing ability, which is a different problem than the second one mentioned above. If an output is near to a prior output, we can think about not only the simplicity of the rather poor neurons of an insect, but also the ability of insect's eye in navigation that supplements its small brain. The information may prove useful to similar researches in neurobiology regarding insect navigation.

6.3 Visual Navigation Using Pixel-wise Information

Chapter 5 presents the third method using pixel-wise landmarks. In this part, we reduce the concept of the landmark into the pixel to find the minimum unit of landmark navigation. Similar to skyline extraction, we check the color distribution on a horizontal line and estimate distribution in each direction by using the ground line. We construct a landmark vector having estimated distance value and check for homing convergence. Then we test another one using only color without distance estimation. We suggest that overall homing pattern is changed by four selections which are perpendicular to each other according to our vector calculation model. We check for one correct answer and convergence of our model. In this part, this model acts like ALV and the four distractors can operate like the order in object matching. Therefore we apply Dynamic Time Warping (DTW) to build an ACV-like holistic model using only color distribution without object detection. It can robustly adjust the sequence of color values in the horizontal line between two images but needs tuning on start and end positions. We add one calculation, based on the flow template of the warping model, locating two stationary points which can be used as focused points to DTW. This refers to the only two points in the panoramic image that are not changed by the motion of the robot. One point is in the approach direction and the other point is in the direction away, which is appropriate to meet the DTW constraint. This DTW was originally developed for finding matching information between time shifted signals and we apply it to visual navigation. In this part, the time shifted information is changed into angular location shifted information and we customize it for navigation by combining two concepts. Through the combination, we can avoid a full search using DTW to find the homing direction and also increase the accuracy of finding the homing direction by landmark vector matching. The overall model can produce successful homing patterns from calculating angular position changes of pixels.

Furthermore, as mentioned earlier, we need to develop the algorithm to operate in complex environments. One solution is the fusion with an image matching method that uses prediction of the image. As we use the omnidirectional image, we can easily change the image in the desired direction without having to move the agent. We can compare the reference image with the current image, then estimate the direction or pose of the robot and we can compensate our model to include these cases. Also, optical flow can be the solution for dynamic situations. There are many optical flow methods such as the one proposed by Lucas Kanade, Srinivasan (Esch et al., 2001; Bouguet, 2001). The basic idea of the optical flow is that the difference in two images and moving objects continuously creates optical flows. We can partially apply it into our model to solve dynamic problems.

Another challenge is the single form of our model using DTW. Through the paper, our model is divided into two methods. One is called wing matching and the other is called unifying matching. Wing matching can reduce overall calculation by 25 percent, but unifying matching produces better output. Each has their own advantages, but they cannot be integrated into one model that takes advantage of their strengths and not their weaknesses. Thus we will evaluate a modified version of our models having to improve performance.

Another area to address is dynamic environments. Based on the snapshot model, we should be able to obtain both simplicity and robustness. However, we did not apply some filtering algorithms for dynamic situations, and therefore we will evaluate a model that uses only background information for finding the current location by filtering out other parts of the environment. Also this attempt may be helpful to our other model using entire pixels that include occlusion.

Additionally, there are experiments in each chapter conducted in both simulations and robot experiments. In this part, we use both handmade environments and open datasets, we also present some mathematical proofs about our models. Based on landmark modeling with certain assumptions, we can partially show the convergence of our navigation model. However, the current model assumes that objects are cylinder-shaped objects and observed distances from one object are almost equal to another from the centers of each mass. In real world, objects are not always cylindrical and distances are measured from their surface, not their center. Therefore we will evaluate our model using general objects of different shapes.

6.4 Future Works

6.4.1 Biological Modeling

As our algorithms originate from the navigation of insects, we used bio-inspired methods. We created a simple artificial retina including various types of receptors. The method using masks can calculate successful the homing direction in a local area. However the method only focused on finding homing direction in the area where overall visual information is certainly sustained. But insects can find the homing direction with occlusion and there is other research regarding animals accomplishing route following over long distances using vision.

We wish to improve our model to be applicable to larger landscapes. Perhaps the answer lies in the discovery of place and grid cell structures, which appear to enable the overall navigation abilities of insects at the neuron level. Firing of each cell has meaning of localization and it must be a super-ordinate concept of local visual navigation.

Through learning phase, we can gather snapshot images in various locations and divide them into small groups based on similarities. Each group is directly connected with place cells like binary codes and also can be one point on the brain map in the grid cell structure. If we create a proper neural network including multi-layers and train it to act like both cells, we can easily find the target snapshot that we want to use at any point along the path. Also, it perhaps makes homing direction in one time.

Different to other cutting-edge deep learning techniques, the assumption would be that the input image has low resolution and the number of neurons actually used in visual navigation must be smaller than in mammals. Therefore we think that the special eye structure can be a clue for visual navigation, similar to our mask-based model. Therefore we can try the simple composition of a neural system with both learning of target images and a learned local snapshot model. Then, after the learning process, our final output can find the homing direction in distant locations as a possible clue of how insect navigation works in the real world.

6.4.2 Engineering Modeling

Our model also takes advantage of engineering modeling by using DTW for pixel matching and, we use the moment function and Haar-like features for information compression. Then we made new mechanisms for finding direction through a combination of those algorithms. However there are limitations that involve navigation in dynamic situations such as having occlusions like illumination changes, objects moving, and location changes. But insects can find the homing direction in dynamic situations and other research indicates that other animals accomplish homing in dynamic situations using vision.

We can consider three areas. First are the changes in the environment from external stimulation, such as changes caused by the location of the sun. Our model without feature matching can partially solve occlusion problem by shape recognition in low resolution and find successful homing direction though not as well as an insect. The fusion of other models with our mask-based model can boost performance to better handle occlusion.

Second is improving the robustness of our models to dynamic environments, which may be best accomplished by applying feature matching to improve the robustness of our moment model. Feature matching can filter out occlusion and one-to-one matching in measurement can be theoretically matched with localization by alignments based on triangulation. Then the accumulation of the information of feature points can be directly connected with both the topological and geographical maps in the learning phase.

Last is the possibility of sensor fusion. Our moment model will gain robustness by using a range sensor without the need for any special algorithms, or adding other types of sensors, such as an odometer.

Appendix A

Appendix

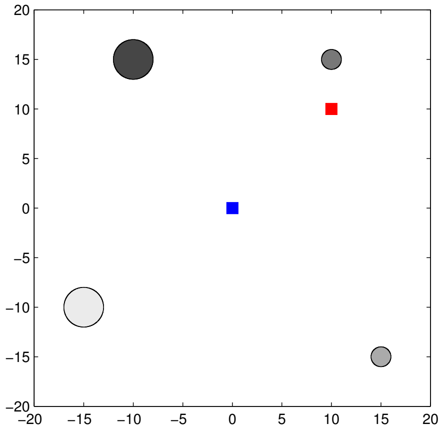
A.1 Environments

In following sections, we use both simulation and robot experiment to check the performances of our methods. The one is simulation with virtual environment with various objects. The other is real environments including various objects.

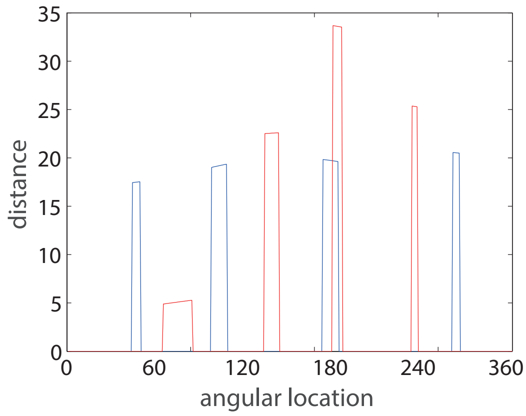
A.1.1 Simulation Environment

In this thesis, we proposed the visual-based landmark navigation. Thus this simulation has imaginary environment including some cylinder-shape objects and it can generate pixel image for horizontal line. Because we only have focus on pixels around horizontal line and this simulation is check for simple action of our models. In this part, we can get visual cues and can test our model without any intervention of disturbances.

Figure A.1 shows the output. It shows the top view of static simulation environment with four cylinder-shape landmarks. Each landmark has position, size and color. The blue square represents the home position and the red square is current position. Next two collected omnidirectional snapshot images include pixels along horizontal line and these pixels' color values are equal to the ones of landmarks. We can change the size of landmark, location, color and measurement positions including occlusion effect. Then we can collect images from simulation to use in homing navigation of our algorithms.



(a)



(b)

Figure A.1: Simulation environment and measured data from the sensors. (a) is top view of simulation environment including four different objects with volume and color. Blue square and red square is measurement positions then measured range data are shown in (b). (c) is expected panoramic image from blue position and (d) from red position.

A.1.2 Experiment Environment

A.1.2.1 Our Dataset

We use robot experiment to check our algorithm in real experiment. This robot has two parts. The one is mobile agent including both mobile and sensor parts. This mobile part is Khphera style robot using two wheels (Roomba, Create 4400, iRobot, Bedford, MA, USA). The other is measurement. There are two types of measurements. The one is visual receptor of omni-directional camera (Logitech Webcam E3500 vision sensor with reflection frame) and the other is range sensor (URG-04LX-UG01, HOKUYO, Osaka, Japan). These two sensors are mounted on mobile part and gathering omni-directional measurements.

Also, those three parts are connected with lap-top computer. This camera sensor col-

(a) (b) (c)

Figure A.2: The part of the mobile robot. (a) i-Robot ROOMBA mobile part, (b) Omnidirectional camera and (c) HOKUYO laser sensor measurement part

lects 720 pixels from horizontal line and laser sensor has about 240 degree region of interest (ROI) with 0.36 degree resolution. Then we have to take 2 shots in two opposite directions to make complete onmi-directional range data.

However, both data from sensors have a lot of numbers and have different resolution. Therefore we reduce them into equal and proper size. First, the visual input has 640 x 480 RGB pixels of omni-directional form. Then we can change them into panorama form by cutting and unfold them according to horizontal line that is virtual flat position with 0.5 degree resolution. Because this horizontal line has almost entire visuo-spatial landmarks in the environment that can be important clue in navigation. As we design this model to local visual homing, this horizontal line have all landmarks when there is no occlusion. After then we can reduce vision input data into 720 pixels in horizontal line.

Second, fitted to converted vision data, we make range data be 720 points. The laser sensor make omnidirectional range data by combining two shots with 1000 numbers of points. Then we can extract the 720 numbers to fit to 0.5 angular resolution which is equal to the visual one. After then, we takes only 720 range points to combine it with visual cues. Then we can get both omni-directional color and depth information with equal resolution as measurement.

With the robot system above, we test several indoor environments for homing navigation. we made test environment called 'ourlab'. It is 5m x 5m empty classroom with various objects like dresser, drawers, trash cans, large vases, windows, walls. We

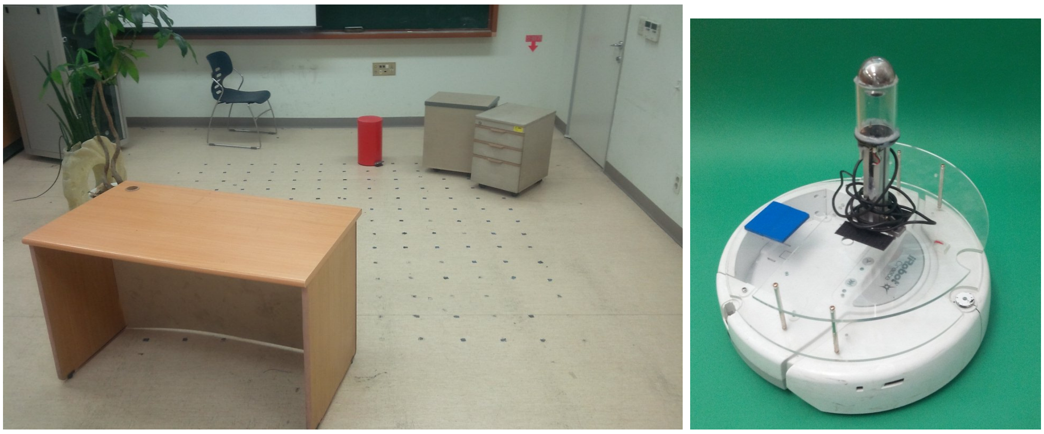


Figure A.3: Omnidirectional images. an image in our lab environment and robot including visual sensor

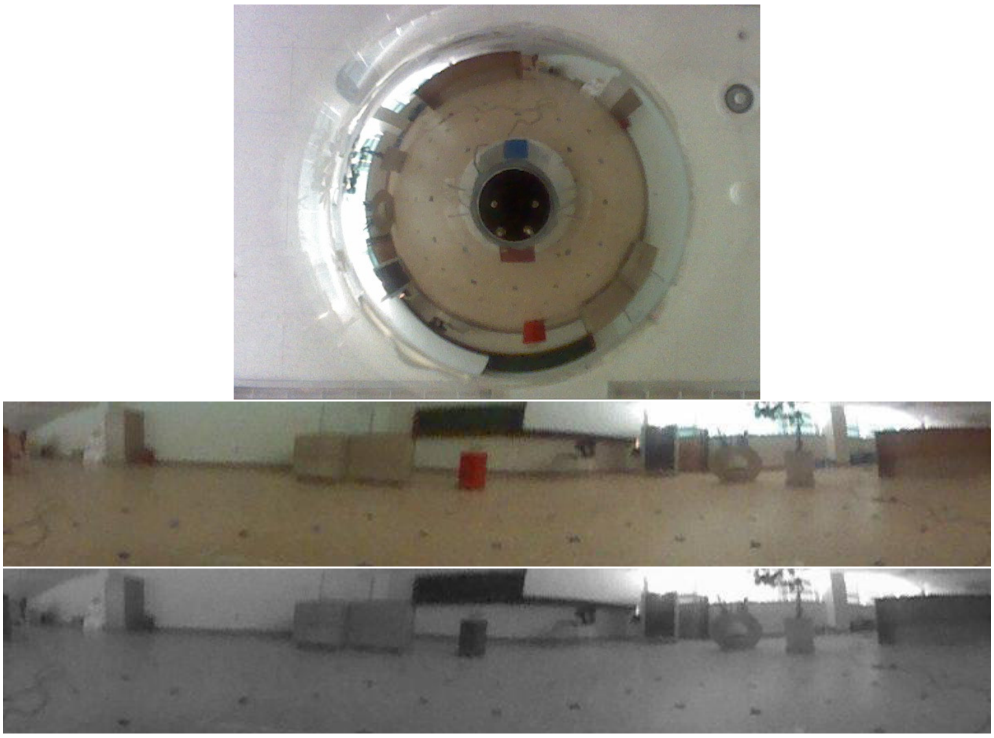


Figure A.4: Environment of Robotic Experiments. First one is a omnidirectional photo. Second one is the example color panoramic image and third one with gray one at center position.

marked the home point and the other measurement points at intervals of 20cm. And then we collect feature measurements on each points to do the experiment. We change the position of these objects to do various experiments. We can see the environment in the left one of Figure A.3.

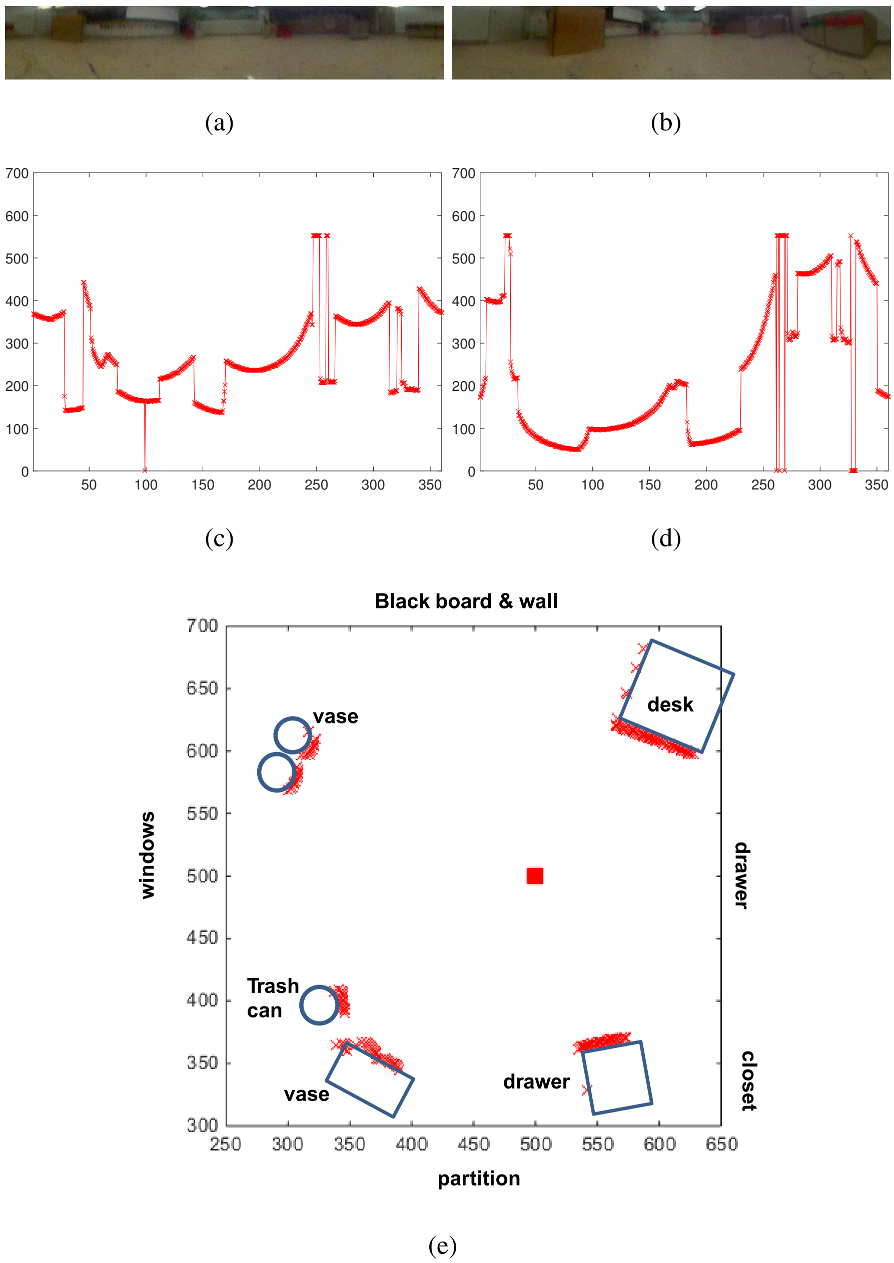


Figure A.5: Data from the sensors. (a) and (b) are the reduced panorama images changed from omni-directional images and (c) and (d) are the reduced laser data. The left ones are taken from location (50,50) and the right ones are from location (62,54). (e) reconstructed reference map by range data with object labels

As the range data has two dimension, we can think these data can be on the horizontal line and we can make reference map including feature points with both position and color information. In Figure A.5, (a) is a picture of our environments including various objects. (b) is captured visual cue at home position (500,500) and (c) is collected range data from equal position. Then reference map (d) shows the reconstructed feature points of environment marked as red x-shapes showing the observed points with name tag. Finally, we make overall measurements into feature points having both RGB pixel values and location from range data. Then we use these data in next experiment.

A.1.2.2 Open Dataset

Addtional to that, we apply our algorithm to common image dataset which is broadly used in local navigation field. We use the one of Andrew Vardy's experiment Vardy and Möller (2005); Möller et al. (2007).

It can be taken from 'http://www.ti.unibielefeld.de/html/research/avardy/'. We use four data. For example, the one called 'A1originalHh (shortly ORIG in our thesis)' is 2.7m x 4.3m and there are 170 points with 10cm intervals. We also change each omnidirectional images into panoramic form like Figure A.6.

Attached to 'ORIG' environment, there are different environments that we use. Others called 'arboreal (ARBO)' and 'screen (SCRE)' are modified version from first (ORIG) one with additional small occlusion. Another called 'hall1 (HALL)' is larger than others with occlusion. Figure A.7 shows the example images from these environments.

We apply our model to this dataset and test our model. We test our model in various ways and check the robustness in performances.

A.2 Moment Model for Homing Navigation

A.2.1 Proof of Global Convergence in Multi-Dimension Case

The general n-th moment model is as follows

$$M_n = \sum_{i=1}^{N} M_{n,i} = \sum_{i=1}^{N} r_i^n C_i = \sum_{i=1}^{N} ((x-a_i)^2 + (y-b_i)^2)^{\frac{n}{2}} C_i \tag{A.1}$$

where we suppose that there is N number of measured landmark. r_i is the range value of i-th landmark which is distance from current location (x,y) to landmark location(a_i,b_i) and C_i is the color value of this landmark as feature.

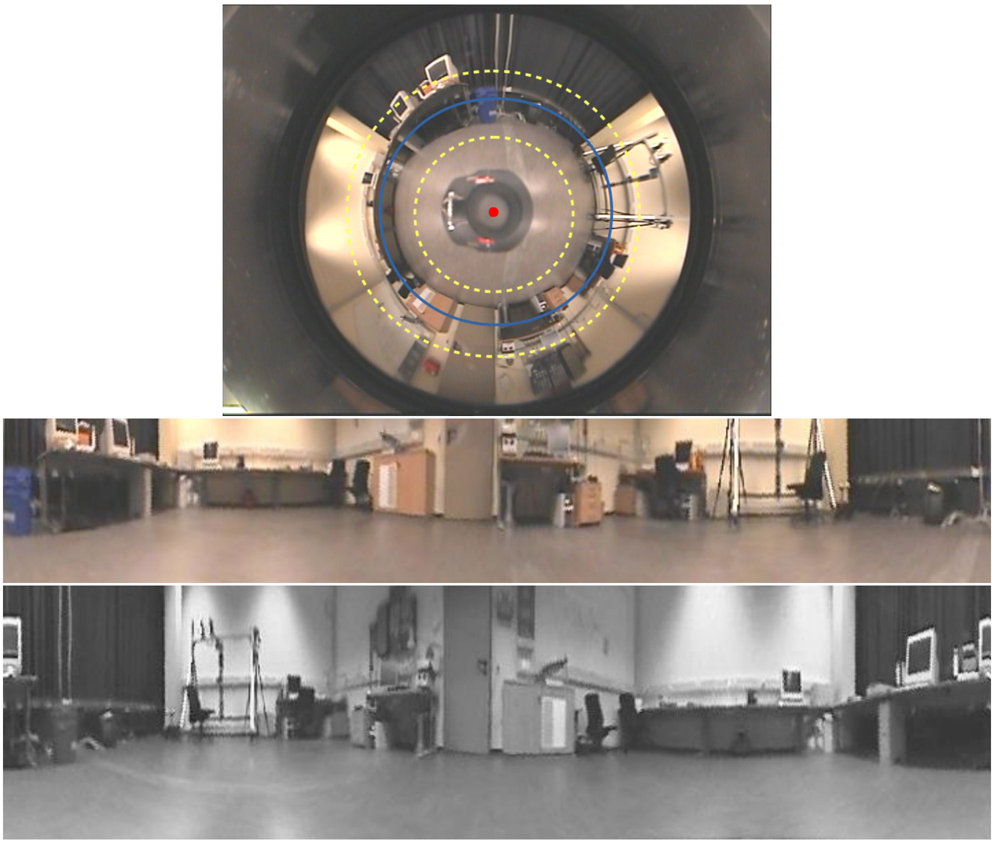


Figure A.6: Example of images used in experiment. First one is omni-directional image that includes other needless parts like outside of mirror. Red dot represents the designated center point and blue line shows the imaginary horizontal line. Dotted yellow lines shows the cutting region. Second one is converted color panoramic form and third one is gray panoramic image

In this part, we try to check the condition of unique convergence point. To solve general solution, we change n variable into $k = n/2$ like

$$k = \frac{n}{2} \rightarrow M_k = \sum_{i=1}^{N} \left((x-a_i)^2 + (y-b_i)^2\right)^k C_i \tag{A.2}$$

Then we take gradient form of moment function to find convexity.

$$\nabla M_k = \sum_{i=1}^{N} k\{(a_i-x)^2 + (b_i-y)^2\}^{k-1}[-2(a_i-x)C_i, -2(b_i-y)C_i] \tag{A.3}$$

where this gradient vector means the changes of potential function to current position (x,y). Above gradient form has first condition about positive order number ($k > 0$). After then, Jacobian matrix has to be calculated.

(a)

(b)

(c)

(d)

Figure A.7: Example images of Vardy's dataset (a) is one of 'A1originalHh (ORIG)' (b) is one of 'screen (SCRE)', (c) of 'arboreal (ARBO)' and (d) of 'hall1 (HALL)'

$$J(\nabla M_k) = \begin{pmatrix} \frac{d^2M}{dx^2} & \frac{d^2M}{dxdy} \\ \frac{d^2M}{dxdy} & \frac{d^2M}{dy^2} \end{pmatrix} = \begin{bmatrix} M_{xx} & M_{xy} \\ M_{yx} & M_{yy} \end{bmatrix} \tag{A.4}$$

where this Jacobian matrix can be calculated by quadratic differential of each term including M_{xx} for quadratic differential about x and M_{yy} about y.

$$= \sum_{i=1}^{N} 2kC_i\{(x-a_i)^2+(y-b_i)^2\}^{k-2} \\ \begin{bmatrix} (2k-1)(x-a_i)^2+(y-b_i)^2 & 2(k-1)(x-a_i)(y-b_i) \\ 2(k-1)(x-a_i)(y-b_i) & (x-a_i)^2+(2k-1)(y-b_i)^2 \end{bmatrix} \tag{A.5}$$

But, this form is too complex to find unique convergence conditions like

$$\det(J(\nabla M_k(X))) > 0 \quad M_{k,xx}(X) > 0 \quad \nabla M_k(X_0) = 0 \tag{A.6}$$

And above three conditions are requirements for function convexity with the unique global convergence to the point X_0. Also, If X_0 can be any real location that is not duplicated with objects, then this model can be the global solution. Therefore we solve above equation.

$$\begin{aligned}
&\alpha_i = 2kC_i\{(x-a_i)^2 + (y-b_i)^2\}^{k-2} \\
&X_i = (x-a_i) \\
&Y_i = (y-b_i) \\
&J(\nabla M_k) = \sum_{i=1}^{N} \alpha_i \begin{bmatrix} (2k-1)X_i^2 + Y_i^2 & 2(k-1)X_iY_i \\ 2(k-1)X_iY_i & X_i^2 + (2k-1)Y_i^2 \end{bmatrix}
\end{aligned} \tag{A.7}$$

First, M_{xx} and M_{yy} are positive with positive $n > 1$. For check the final condition, determinant output of multidimensional moment function has to be calculated.

$$\begin{aligned}
\det(J(\nabla M_k)) = &\sum_{i=1}^{N} \alpha_i\{(2k-1)X_i^2 + Y_i^2\} \sum_{j=1}^{N} \alpha_j\{X_j^2 + (2k-1)Y_j^2\} \\
&- \sum_{i=1}^{N} \alpha_i\{2(k-1)X_iY_i\} \sum_{j=1}^{N} \alpha_j\{2(k-1)X_jY_j\}
\end{aligned} \tag{A.8}$$

In this part, we divide them into two terms. The one is $i = j$ and the other is $i \neq j$ that is marked as Λ.

$$\begin{aligned}
&= \sum_{i=1}^{N} [\alpha_i^2(2k-1)(X_i^2 + Y_i^2)^2] + \Lambda \\
&= \sum_{i=1}^{N} [(2kC_i\{(a_i-x)^2 + (b_i-y)^2\}^{k-2})^2(2k-1)\{(a_i-x)^2 + (b_i-y)^2\}^2] + \Lambda \\
&= \sum_{i=1}^{N} 4C_i^2k^2(2k-1)\{(a_i-x)^2 + (b_i-y)^2\}^{2k-2} + \Lambda
\end{aligned} \tag{A.9}$$

we return the variable k into $n(n = 2k)$

$$\begin{aligned}
&= \sum_{i=1}^{N} C_i^2 n^2(n-1) r_i^{2n-4} + \Lambda \\
&= \sum_{i=1}^{N} n^2(n-1)M_{i,n}^2/r_i^4 + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \alpha_i\alpha_j[\{(n-1)^2+1\}(X_iY_j - X_jY_i)^2 + 2(n-1)(X_iX_j + Y_iY_j)^2
\end{aligned}$$

Because detail solution procedure of Λ is

$$\begin{aligned}
\Lambda &= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\alpha_i\alpha_j\left(\begin{array}{c}(n-1)X_i^2X_j^2+(n-1)^2X_i^2Y_j^2+X_j^2Y_i^2+(n-1)Y_i^2Y_j^2\\ +(n-1)X_j^2X_i^2+(n-1)^2X_j^2Y_i^2+X_i^2Y_j^2+(n-1)Y_j^2Y_i^2\\ -2(n-2)^2X_iX_jY_iY_j\end{array}\right)\\
&= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\alpha_i\alpha_j[(n^2-2n+2)(X_iY_j-X_jY_i)^2+2(n-1)(X_iX_j+Y_iY_j)^2]\\
&= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\alpha_i\alpha_j[\{(n-1)^2+1\}(X_iY_j-X_jY_i)^2+2(n-1)(X_iX_j+Y_iY_j)^2]
\end{aligned} \tag{A.10}$$

Then final output can be written as

$$f\left[\begin{array}{ccc} n\geq 1 & \det(J(\nabla M_k(X)))>0 & M_{k,xx}(X)>0\\ 0<n<1 & \mathit{unknown} & M_{k,xx}(X)>0\\ n\leq 0 & \mathit{unknown} & M_{k,xx}(X)<0\end{array}\right. \tag{A.11}$$

Then, we can check that there must be unique convergence point with $n \geq 1$ and we can find the right homing direction when we use $n \geq 1$. Different to the prior one, the case with $n \leq 0$ has no convergence point and the other case with $0 < n < 1$ can not be sure of the existence of unique convergence point.

A.2.2 Landmark Modelling

Theorem 1. As there is objects in the field with cylindrical shapes without occlusion, then the summation of range vectors pointing angles within those objects can be represented by

$$\overrightarrow{O} = \sum_i \eta\frac{R_i}{r_i}\overrightarrow{u}(\theta_i) \tag{A.12}$$

where $\overrightarrow{u}(\theta_i)$ is unit vector pointing to particular angle(θ_i) and R_i is radius of the object and r_i is distance between view point and center of the object. η is the normalize constant which is affected by angular resolution.

Proof: In this part, we explain our landmark modelling method. The landmark is a entirely noticed object (in experiment area) that can be considered as geometrical feature in navigation. We apply the idea of pixel-wise landmark which means that we

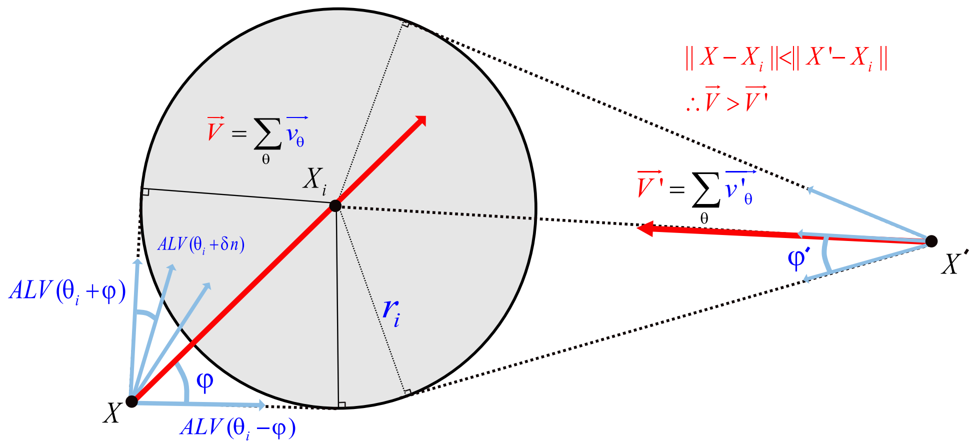


Figure A.8: Principle and concept of pixel-wise landmark. Overall figure is top-view form. Gray circle means cylinder shape landmark and X_i is the center with radius r_i. There are two observation at X and X'. Then equal distance vectors called ALV point to part of object that are marked as blue arrows. The sum of blue ones is red arrow that is not geometrically equal to real landmark vector.

treat pixel as landmark itself without clustering algorithm to interpret surrounding environment. In this part, we put one assumption that all the objects have cylinder shape. Because this cylinder shape is not affected by pose of the robot in horizontal direction. We think if this modelling is right, then other shapes are also be approximated. When there is only object (not background), final summation of pixel-wise landmark vector is $\vec{O}$ and it can be written as

$$\vec{O} = \sum_i \vec{V_i} \simeq \sum_i \lim_{\delta \to 0} \vec{V_i} = \sum_i \eta \frac{R_i}{r_i} \vec{u}(\theta_i) \tag{A.13}$$

where, refer to Figure A.8, $\vec{u}(\theta_i)$ is unit length vector point to the center of i-th landmark object and $\vec{u}(\theta_i)$ is the perpendicular unit vector component of $\vec{u}(\theta_i)$ then $\vec{u}(\theta_i) \cdot \vec{w}(\theta_i) = 0$. $r_i(X)$ is distance value between position X and landmark location X_i and size(radius) of object is r_i. δ is angular resolution and $\varphi(= \sin^{-1}(\frac{R_i}{||X_i - X||}))$ is angle of right triangle including both radius and distance.

Because

$$\begin{aligned}
&\overrightarrow{V_i} = \sum_{\theta=-\varphi}^{\varphi} \overrightarrow{u}(\theta_i+\theta) = \overrightarrow{u}(\theta_i)\{\cos\varphi + \cos(\varphi-\delta) + \cos(\varphi-2\delta)\cdots + \cos(-\varphi)\} \\
&+\overrightarrow{w}(\theta_i)\{\sin\varphi + \sin(\varphi-\delta) + \sin(\varphi-2\delta)\cdots + \sin(-\varphi)\} \\
&\lim_{\delta\to 0}\overrightarrow{V_i} = \overrightarrow{u}(\theta_i)\eta\int_{-\varphi}^{\varphi}\cos\theta d\theta + \overrightarrow{w}(\theta_i)\eta\int_{-\varphi}^{\varphi}\sin\theta d\theta = \eta\overrightarrow{u}(\theta_i)\times 2\sin\varphi + 0 \\
&= \eta\overrightarrow{u}(\theta_i)\times 2\sin(\sin^{-1}(\tfrac{R_i}{||X_i-X||})) \\
&= \eta\tfrac{2R_i}{r_i}\overrightarrow{u}(\theta_i)
\end{aligned} \tag{A.14}$$

where normalizer (η) is about the constant for angular resolution. In real part, we use angular resolution as normalizer and use it into modelling output.

A.2.3 Applying Modelling into Navigation Model

Theorem 2. As there is objects in the field with cylindrical shapes without occlusion, based on the **Theorem 1**, then adding pixel-wise landmark vectors to the square of each distance can operate like extracting objects without an object extraction algorithm.

$$\overrightarrow{H}(x,y) = -\nabla M = \sum_i w_i r_i^2 \overrightarrow{\theta_i} - w_{i0} r_{i0}^2 \overrightarrow{\theta_{i0}} = \sum_i r_i C_i[a_i - x, b_i - y] - r_i C_i[a_i - x_0, b_i - y_0] \tag{A.15}$$

where $\overrightarrow{H}(x,y)$ is calculated homing direction and (x,y) is current location and (x_0,y_0) is home position. r_{i0} is distance between home and i-th measurement and r_i between current and i-th measurement. θ_{i0} is pointing angular position to measurement from home and θ_i is from current position. i is i-th object in this part and (a_i,b_i) is center location of measurement in reference map. R_i is radius and C_i is measured feature value of this object. Then output is approximated to our original moment model despite of using pixel-wise landmarks.

Proof: In this part, we try to apply pixel-wise landmark modelling into holistic matching model.

$$\nabla M = \sum_{\theta=0}^{360^\circ} \overrightarrow{r_0}(\theta) - \overrightarrow{r}(\theta) = \sum_{i=1}^{N} w_{i0} r_{i0} \overrightarrow{\theta_{i0}} - w_i r_i \overrightarrow{\theta_i} \tag{A.16}$$

The equation above is modelling of omnidirectional range data without object extraction. Then we can get both range and direction. With pixel-wise landmark model, these omnidirectional ranges can be changed into the weighted form of landmark model. Each weight is composed of the summation about landmark vectors of equal object and it is inversely proportional to distance to this object. In this case, we think $r_i >> R_i$ and the environment without occlusion to simply approximate navigation model. The output can be represented by

$$\begin{aligned}
&\nabla M = \sum_i w_{i0} r_{i0} \overrightarrow{\theta_{i0}} - w_i r_i \overrightarrow{\theta_i} \\
&= \eta \sum_i \left(\tfrac{R_i C_i}{||X_i - X_0||}\right)[a_i - x_0, b_i - y_0] - \left(\tfrac{R_i C_i}{||X_i - X||}\right)[a_i - x, b_i - y] \\
&M_{xx} = \eta \sum_i R_i C_i \left(-\tfrac{(x-a_i)^2}{||X_i - X||} + ||X_i - X||\right) = \eta \sum_i R_i C_i \tfrac{(y-b_i)^2}{||X_i - X||} \\
&M_{yy} = \eta \sum_i R_i C_i \left(-\tfrac{(y-b_i)^2}{||X_i - X||} + ||X_i - X||\right) = \eta \sum_i R_i C_i \tfrac{(x-a_i)^2}{||X_i - X||} \\
&M_{xy} = \eta \sum_i -R_i C_i \tfrac{(x-a_i)(y-b_i)}{||X_i - X||} \\
&\lambda_i = ||\overrightarrow{X}_i - \overrightarrow{X}|| = ||(X_i, Y_i)|| = ||(a_i - x), (b_i - y)|| \\
&D = M_{xx} M_{yy} - M_{xy}^2 = \eta^2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \tfrac{(X_i Y_j - X_j Y_i)^2}{\lambda_i \lambda_j} > 0 \\
&\exists i, j (i \neq j) : (b_i - y)(x_j - x) - (a_i - x)(y_j - y) \neq 0 \\
&M_{xx}(X) > 0, \nabla M(X_0) = 0, D > 0
\end{aligned} \tag{A.17}$$

where i and j are the indexes of object, R for radius, C for color intensity, (a_i, b_i) for the center location of i-th object, $\overrightarrow{\theta}$ for unit vector pointing direction θ and

$$J(\nabla M_k) = \begin{pmatrix} \frac{d^2 M}{dx^2} & \frac{d^2 M}{dxdy} \\ \frac{d^2 M}{dxdy} & \frac{d^2 M}{dy^2} \end{pmatrix} = \begin{bmatrix} M_{xx} & M_{xy} \\ M_{yx} & M_{yy} \end{bmatrix} \tag{A.18}$$

this Jacobian matrix can be calculated by quadratic differential of each term including M_{xx} for quadratic differential about x and M_{yy} about y.

But the output is different to our object like

$$\begin{aligned}
&\nabla M = \sum_i \left(\tfrac{R_i C_i}{||X_i - X_0||}\right)[a_i - x_0, b_i - y_0] - \left(\tfrac{R_i C_i}{||X_i - X||}\right)[a_i - x, b_i - y] \\
&\neq \sum_i 2C_i [a_i - x_0, b_i - y_0] - 2C_i [a_i - x, b_i - y]
\end{aligned} \tag{A.19}$$

Therefore we apply r_i^2 to remove $1/r$ part of gradient component.

$$\begin{aligned}
&\nabla M = \sum_i w_{i0} r_{i0}^2 \overrightarrow{\theta_{i0}} - w_i r_i^2 \overrightarrow{\theta_i} = \sum_i R_i C_i [a_i - x_0, b_i - y_0] - R_i C_i [a_i - x, b_i - y] \\
&M_{xx} = \sum_i R_i C_i = M_{yy} \\
&M_{xy} = 0 \\
&D = M_{xx} M_{yy} - M_{xy}^2 > 0 \\
&M_{xx}(X) > 0, D(X) > 0, \nabla M(X_0) = 0
\end{aligned} \tag{A.20}$$

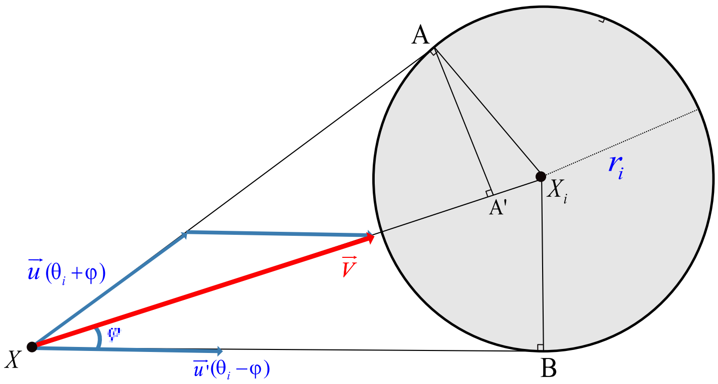


Figure A.9: Top view illustrating the landmark model. The gray circle represents an object with cylindrical shape of radius r_i and located at position X_i, whereas X represents the position of observation. The model can robustly detect the landmark edges and return unit vectors (blue arrows). The red arrow represents the sum of the two unit vectors.

where X_0 is home position In short, our model in the case using pixel-wise landmarks has to use r_i^2 instead of r_i because the overall summation is fit with original our model.

A.3 Visual Mask Model for Homing Navigation

A.3.1 Landmark Modelling

To perform visual homing navigation, we represent objects as cylinders to interpret the surrounding environment. For the landmark model, we determine the sum of all edges assuming no occlusions. Then, we can identify individual objects, as illustrated in Figure A.9. We consider two unit vectors per object and use their sum in the proposed method. The generated vector, $\vec{V}$, is different to the real vector for home position (X) to object center (X_i), but there is a relation between them.

$$r_i = ||X_i - X||$$
$$\angle X_iXA = \angle AXA' = \angle X_iXB = \varphi = \sin^{-1}(\frac{R_i}{r_i})$$

where r_i is radius and D_i is distance of i-th object from object location (X_i) to current location (X). A mathematical description based on the above relations allows to determine the vector length as

$$\Lambda = ||\overrightarrow{V}|| = ||\overrightarrow{u}(\theta_i + \varphi) + \overrightarrow{u}(\theta_i - \varphi)||$$

where Λ is a length of the generated vector (V) and weight (w_i) can be presented by

$$w_i = 2\frac{\Lambda}{r_i} = 2\frac{\{r_i \cos(\varphi)\}\cos(\varphi)}{r_i} = 2[\cos\{\sin^{-1}(\frac{R_i}{r_i})\}]^2 = 2(1 - \frac{R_i^2}{r_i^2}).$$

A.3.2 Proof of Global Convergence of Homing Vector

The proposed method is designed to find the homing direction by comparing two representative vectors corresponding to visual cues.

$$\overrightarrow{H} = -\nabla M = \overrightarrow{Curr} - \overrightarrow{Home}$$
$$\nabla M = \sum_{\theta} \delta(I_0, \theta)\overrightarrow{u_{\theta 0}} - \delta(I, \theta)\overrightarrow{u_{\theta}} = \sum_i w_{i0}\overrightarrow{u_{i0}} - \sum_i w_i \overrightarrow{u_i}$$

where

$$\delta(I, \theta) = \begin{bmatrix} 1, & if\_I(\theta) = edge \\ 0, & otherwise \end{bmatrix}$$

.

Using the landmark model, the above vector summation is simplified by using the edge finding function Then, the gradient can be written as

$$\nabla M = \sum_i 2(1 - \frac{R_i^2}{r_{i0}^2})X_0 - 2(1 - \frac{R_i^2}{r_i^2})X$$

where R_i is radius and r_i is distance of i-th object from object location to current location ($X = (x, y)$). r_0 is distance from object location to home location ($X_0 = (x_0, y_0)$). If the gradient is convex with a global minimum, then our the homing vector can reach home. In this situation, the required conditions are

$$M_{xx}(X) > 0, \nabla M(X_0) = 0, D(X) > 0$$

where D is determinant of Hessian matrix and

$$J(\nabla M_k) = \begin{pmatrix} \frac{d^2M}{dx^2} & \frac{d^2M}{dxdy} \\ \frac{d^2M}{dxdy} & \frac{d^2M}{dy^2} \end{pmatrix} = \begin{bmatrix} M_{xx} & M_{xy} \\ M_{yx} & M_{yy} \end{bmatrix}$$

Jacobian matrix can be calculated by quadratic differential of each term including M_{xx} for quadratic differential about x and M_{yy} about y.

Next, we calculate the conditions for global convergence:

$$
\begin{aligned}
&M_{xx} = \sum_i \frac{\partial(-2(1-\frac{R_i^2}{r_i^2})X)}{\partial x} = \sum_i \frac{1}{r_i^5}(Y_i^4 + (X_i^2 - R_i^2)Y_i^2 + 2X_i^2R_i^2) \quad \leftarrow r_i \geq R_i, \\
&M_{xx} \geq \sum_i \frac{1}{r_i^5}(Y_i^4 + (X_i^2 - r_i^2)Y_i^2 + 2X_i^2R_i^2) = \sum_i \frac{1}{r_i^5}(2X_i^2r_i^2) > 0, \\
&M_{yy} \geq \sum_i \frac{1}{r_i^5}(X_i^4 + (Y_i^2 - r_i^2)X_i^2 + 2Y_i^2R_i^2) = \sum_i \frac{1}{r_i^5}(2Y_i^2R_i^2) > 0, \\
&M_{xy} = \sum_i \frac{\partial(-2(1-\frac{R_i^2}{r_i^2})X)}{\partial y} = \sum_i \frac{1}{r_i^5}(X_iY_i(3R_i^2 - r_i^2)) \leftarrow r_i \geq R_i, \\
&M_{xy} \leq \sum_i \frac{1}{r_i^5}(2X_iY_ir_i^2).
\end{aligned}
$$

where $X_i = x_i - x$ and $Y_i = y_i - y$, R_i for radius and r_i for distance from current location $(X = (x, y))$. Then, we can verify the determinant of the Jacobian:

$$
\begin{aligned}
&D(X) = M_{xx}M_{yy} - (M_{xy})^2 \geq \sum_i \sum_j \frac{2}{r_i^5}\frac{2}{r_j^5}\{(X_i^2R_i^2)(Y_j^2R_j^2) - (X_iY_ir_i^2)(X_jY_jr_j^2)\} \leftarrow r_i \geq R_i \\
&= \sum_{i=1}^{N} 4X_i^2Y_i^2(R_i^4 - r_i^4) + 4\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\left[\begin{array}{c} X_i^2Y_j^2R_i^2R_j^2 - X_iY_iX_jY_jr_i^2r_j^2 \\ +X_j^2Y_i^2R_j^2R_i^2 - X_iY_iX_jY_jr_i^2r_j^2 \end{array}\right] \\
&= 0 + 4\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\{(X_iY_j - X_jY_i)^2r_i^2r_j^2\} \geq 0.
\end{aligned}
$$

Therefore, we obtain two conditions that are required to prove global convergence:

$$
\exists i, j (i \neq j) : (y_i - y)(x_j - x) - (x_i - x)(y_j - y) \neq 0, \quad r_i \geq R_i.
$$

Furthermore, we can identify cases where converge cannot be satisfied. For instance, Figure A.10 shows a case that can contradict converge. At the home position, the radius and distance to the center of the objects are equal, and they are aligned and placed together. In this situation, homing navigation will mostly fail.

Nevertheless, such situations are less severe than those occurring when using techniques such as the ALV method, Thus, we consider that the demonstrated convergence is, in general, a good feature of the proposed method.

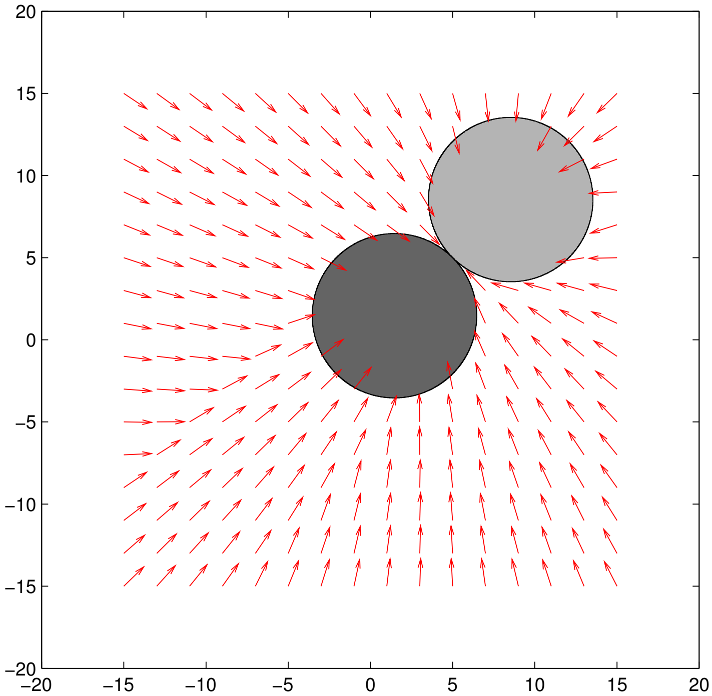


Figure A.10: Case where $D(J(\nabla M)) = 0$. The red square and arrows represent the home position and homing directions with an angular resolution of 1°. The gray circles representing objects have equal radius, which is also equal to the distance between the home position and the object centers. In addition, the objects are aligned through the home position, and hence $D(J(\nabla M))$ can be zero at this position.

A.4 Pixel-Wise Landmark Model for Homing Navigation

A.4.1 Landmark Modelling

As our assumption is that there are perfectly cylinder shape landmarks. When there is only object (not background), final summation of pixel-wise landmark vector is $\overrightarrow{O}$ and it can be write

$$\overrightarrow{O} = \sum_i \overrightarrow{V_i} \simeq \sum_i \lim_{\delta \to 0} \overrightarrow{V_i} = \sum_i \frac{R_i}{r_i} \overrightarrow{u}(\theta_i) \tag{A.21}$$

where, refer to Figure 5.2, $\overrightarrow{u}(\theta_i)$ is unit length vector point to the center of i-th landmark object and $\overrightarrow{u}(\theta_i)$ is the perpendicular unit vector component of $\overrightarrow{u}(\theta_i)$ then $\overrightarrow{u}(\theta_i) \cdot \overrightarrow{w}(\theta_i) = 0$. $r_i(X)$ is distance value between position X and landmark location X_i and size(radius) of object is R_i. δ is angular resolution and $\varphi(= \sin^{-1}(\frac{R_i}{||X_i - X||}))$ is angle of right triangle including both radius and distance. Because

$$\begin{aligned}
&\overrightarrow{V_i} = \sum_{\theta=-\varphi}^{\varphi} \overrightarrow{u}(\theta_i+\theta) = \overrightarrow{u}(\theta_i)\{\cos\varphi+\cos(\varphi-\delta)+\cos(\varphi-2\delta)\cdots+\cos(-\varphi)\} \\
&+\overrightarrow{w}(\theta_i)\{\sin\varphi+\sin(\varphi-\delta)+\sin(\varphi-2\delta)\cdots+\sin(-\varphi)\} \\
&\lim_{\delta\to 0}\overrightarrow{V_i} = \overrightarrow{u}(\theta_i)\eta\int_{-\varphi}^{\varphi}\cos\theta d\theta+\overrightarrow{w}(\theta_i)\eta\int_{-\varphi}^{\varphi}\sin\theta d\theta = \eta\overrightarrow{u}(\theta_i)\times 2\sin\varphi+0 \\
&= \eta\overrightarrow{u}(\theta_i)\times 2\sin(\sin^{-1}(\tfrac{R_i}{||X_i-X||})) \\
&= \eta\tfrac{2R_i}{r_i}\overrightarrow{u}(\theta_i)
\end{aligned} \tag{A.22}$$

In this part, normalizer (η) is about the term of angular resolution. In real part, we remove it and use modelling output.

A.4.2 Proof of Homing Convergence with Distance Information

To prove unique convergence of our homing model using distance information, suggested model has to satisfy three conditions. These are

$$M_{xx}(X) > 0, D(X) > 0, \nabla M(X_0) = 0 \tag{A.23}$$

where

$$J(\nabla M_k) = \begin{pmatrix} \frac{d^2M}{dx^2} & \frac{d^2M}{dxdy} \\ \frac{d^2M}{dxdy} & \frac{d^2M}{dy^2} \end{pmatrix} = \begin{bmatrix} M_{xx} & M_{xy} \\ M_{yx} & M_{yy} \end{bmatrix} \tag{A.24}$$

where this Jacobian matrix can be calculated by quadratic differential of each term including M_{xx} for quadratic differential about x and M_{yy} about y. Attached with modeling above, if r_i is larger than R_i, homing vector (∇M) can be approximately presented like

$$\begin{aligned}
&\nabla M = \sum_i w_{i0}r_{i0}\overrightarrow{\theta_{i0}} - w_i r_i\overrightarrow{\theta_i} = \sum_i (\tfrac{R_iC_i}{||X_i-X_0||})[a_i-x_0, b_i-y_0] - (\tfrac{R_iC_i}{||X_i-X||})[a_i-x, b_i-y] \\
&M_{xx} = \sum_i R_iC_i(-\tfrac{(x-a_i)^2}{||X_i-X||}+||X_i-X||) = \sum_i R_iC_i\tfrac{(y-b_i)^2}{||X_i-X||} > 0 \\
&M_{yy} = \sum_i R_iC_i(-\tfrac{(y-b_i)^2}{||X_i-X||}+||X_i-X||) = \sum_i R_iC_i\tfrac{(x-a_i)^2}{||X_i-X||} \\
&M_{xy} = \sum_i -R_iC_i\tfrac{(x-a_i)(y-b_i)}{||X_i-X||}
\end{aligned} \tag{A.25}$$

where R_i for radius, C_i for color intensity, r_i for distance from current location ($X = (x,y)$) to i-th object location ($X_i = (a_i, b_i)$) and r_{i0} for distance from home location ($X_0 = (x_0, y_0)$). Then determinant of Jacobian condition can be calculated by

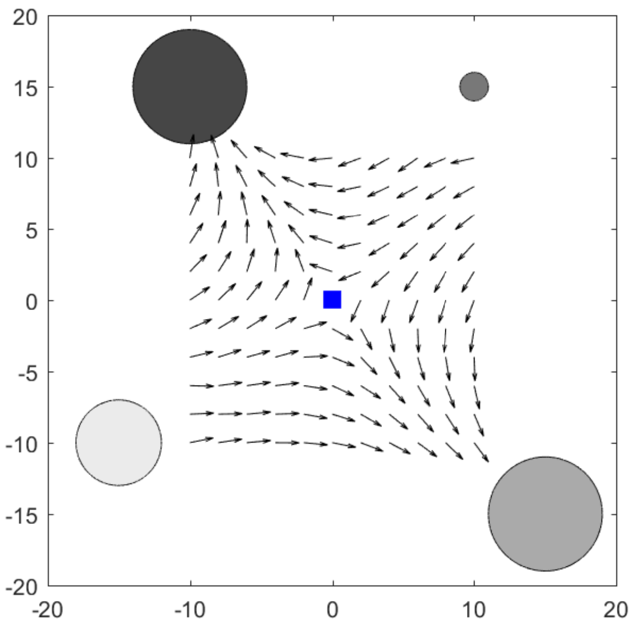


Figure A.11: Counterexample of model using (X_F, Y_F) in homing navigation. There are four landmarks with home position (0,0) and black arrows show the calculated homing direction by using (X_F, Y_F).

$$\begin{aligned} &\lambda_i = ||\overrightarrow{X}_i - \overrightarrow{X}|| = ||(X_i, Y_i)|| = ||(a_i - x), (b_i - y)|| \\ &D = M_{xx}M_{yy} - M_{xy}^2 = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{(X_iY_j - X_jY_i)^2}{\lambda_i \lambda_j} > 0 \end{aligned} \tag{A.26}$$

with one criteria

$$\exists i, j (i \neq j) : (b_i - y)(a_j - x) - (a_i - x)(b_j - y) \neq 0 \tag{A.27}$$

A.4.3 Proof of Homing Convergence without Distance Information

To prove the unique convergence of our homing model without distance information, the suggested model showed to satisfy three conditions.

$$M_{xx}(X) > 0, det(J(\nabla M(X))) > 0, \nabla M(X_0) = 0 \tag{A.28}$$

where

$$J(\nabla M) = \begin{pmatrix} \frac{d^2M}{dx^2} & \frac{d^2M}{dxdy} \\ \frac{d^2M}{dxdy} & \frac{d^2M}{dy^2} \end{pmatrix} = \begin{bmatrix} M_{xx} & M_{xy} \\ M_{yx} & M_{yy} \end{bmatrix} \tag{A.29}$$

This Jacobian matrix can be calculated by quadratic differential of each term including M_{xx} for quadratic differential about x and M_{yy} about y. Attached with modelling without distance, if r_i is larger than R_i, the potential field (M) of homing vector ($-\nabla M$) can be given by

$$M = \sum_i [R_i C_i \{\ln(||X_i - X||^2) - \frac{1}{||X_i - X_0||^2} X_0 \cdot X\}] \tag{A.30}$$

$$\nabla M = \sum_i (\frac{\eta R_i C_i}{||X_i - X_0||^2})[a_i - x_0, b_i - y_0] - (\frac{\eta R_i C_i}{||X_i - X||^2})[a_i - x, b_i - y] \tag{A.31}$$

where, for i-th landmark, R_i for radius, C_i for color intensity, $r_i = ||X_i - X||$ for distance from the current location ($X = (x,y)$) to the object location ($X_i = (a_i, b_i)$) and $r_{i0} = ||X_i - X_0||$ for distance from the home location ($X_0 = (x_0, y_0)$). Then, homing vector($-\nabla M$) is

$$\overrightarrow{H} = -\nabla M = \overrightarrow{O(X)} - \overrightarrow{O(X_0)} = (X_F, Y_F) \tag{A.32}$$

where (X_F, Y_F) is directly calculated homing direction using potential. However, in this part, the output cannot be converged to home position (X_0). Because it does not satisfy the one of the unique convergence conditions ($D(X) > 0$).

$$\begin{aligned}
&\nabla M = \eta \sum_i (\tfrac{R_i C_i}{||X_i - X_0||^2})[a_i - x_0, b_i - y_0] - (\tfrac{R_i C_i}{||X_i - X||^2})[a_i - x, b_i - y] \\
&M_{xx} = \eta \sum_i -R_i C_i \tfrac{(x-a_i)^2 - (y-b_i)^2}{||X_i - X||^4} = M_{yy} \\
&M_{xy} = \eta \sum_i 2 R_i C_i \tfrac{(x-a_i)(y-b_i)}{||X_i - X||^4} = M_{yx} \\
&det(J(\nabla M)) = M_{xx} M_{yy} - M_{xy} M_{yx} = (M_{xx})^2 - (M_{xy})^2
\end{aligned} \tag{A.33}$$

Figure A.11 shows the failed counter example using X_F and Y_F as they are. The above potential M does not satisfy one of convergence conditions, $det(J(\nabla M)) > 0$, and cannot guide homing. Thus, new gradient of potential fields to support the convergence are suggested. We propose four homing vectors ($\nabla M_1, \nabla M_2, \nabla M_3, \nabla M_4$) as given by:

$$\begin{aligned}
\nabla M_1 &= -\nabla M_2 = (X_F, -Y_F) \\
\nabla M_2 &= -\nabla M_1 = (-X_F, Y_F) \\
\nabla M_3 &= -\nabla M_4 = (Y_F, X_F) \\
\nabla M_4 &= -\nabla M_3 = (-Y_F, -X_F)
\end{aligned} \tag{A.34}$$

where ∇M_1, ∇M_2, ∇M_3 and ∇M_4 are extracted from ∇M which is not successful without distance information. These terms do not use simple X_F and Y_F; instead, they change their sign or order. Two different types are perpendicular to each other and two different direction is opposite ($\nabla M_1 \cdot \nabla M_2 = 0, \nabla M^+ = -\nabla M^-$). A sequence of a homing vector produces a homing route what homing vectors guide homing should be investigated. Then those four gradient functions are given by

$$
\begin{aligned}
\nabla M_1 &= \sum_i \left(\tfrac{\eta R_i C_i}{||X_i - X_0||^2}\right)[a_i - x_0, -b_i + y_0] - \left(\tfrac{\eta R_i C_i}{||X_i - X||^2}\right)[a_i - x, -b_i + y] \\
\nabla M_2 &= \sum_i \left(\tfrac{\eta R_i C_i}{||X_i - X_0||^2}\right)[-a_i + x_0, b_i - y_0] - \left(\tfrac{\eta R_i C_i}{||X_i - X||^2}\right)[-a_i + x, b_i - y] \\
\nabla M_3 &= \sum_i \left(\tfrac{\eta R_i C_i}{||X_i - X_0||^2}\right)[b_i - y_0, a_i - x_0] - \left(\tfrac{\eta R_i C_i}{||X_i - X||^2}\right)[b_i - y, a_i - x] \\
\nabla M_4 &= \sum_i \left(\tfrac{\eta R_i C_i}{||X_i - X_0||^2}\right)[-b_i + y_0, -a_i + x_0] - \left(\tfrac{\eta R_i C_i}{||X_i - X||^2}\right)[-b_i + y, -a_i + x]
\end{aligned}
\tag{A.35}
$$

where C_i is feature value (color) and R_i is radius of i-th landmark at location ($X_i = (a_i, b_i)$) from current location ($X = (x, y)$) and home location ($X_0 = (x_0, y_0)$). Two different types of suggested models are perpendicular to each other, and two different directions are opposite ($\nabla M_1 \cdot \nabla M_3 = 0, \nabla M_1 = -\nabla M_2, \nabla M_3 = -\nabla M_4$). We try to observe the convergence conditions of these four models. To solve the convexity, we calculate quadratic differentials like

$$
\begin{aligned}
M_{1,xx} = -M_{4,xy} &= \sum_i -\eta R_i C_i \tfrac{(x_0 - a_i)^2 - (y_0 - b_i)^2}{||X_i - X_0||^4} = \sum_i -\eta R_i C_i \cos(2\theta_i)/D_i^2 \\
M_{2,xx} = -M_{1,xx} &= \sum_i \eta R_i C_i \tfrac{(x_0 - a_i)^2 - (y_0 - b_i)^2}{||X_i - X_0||^4} = \sum_i \eta R_i C_i \cos(2\theta_i)/D_i^2 \\
M_{3,xx} = -M_{2,xy} &= \sum_i -\eta R_i C_i \tfrac{2(x_0 - a_i)(y_0 - b_i)}{||X_i - X_0||^4} = \sum_i -\eta R_i C_i \sin(2\theta_i)/D_i^2 \\
M_{4,xx} = -M_{3,xx} &= \sum_i \eta R_i C_i \tfrac{2(x_0 - a_i)(y_0 - b_i)}{||X_i - X_0||^4} = \sum_i \eta R_i C_i \sin(2\theta_i)/D_i^2
\end{aligned}
\tag{A.36}
$$

Using those outputs, we observe the convergence conditions of these four models. First,

$$
\begin{aligned}
&\nabla M_1 = \eta \sum_i \left(\tfrac{R_i C_i}{||X_i - X_0||^2}\right)[a_i - x_0, -b_i + y_0] - \left(\tfrac{R_i C_i}{||X_i - X||^2}\right)[a_i - x, -b_i + y] \\
&M_{xx} = \eta \sum_i -R_i C_i \tfrac{(x - a_i)^2 - (y - b_i)^2}{||X_i - X||^4} = M_{yy} \\
&M_{xy} = \eta \sum_i 2 R_i C_i \tfrac{(x - a_i)(-y + b_i)}{||X_i - X||^4} = -M_{yx} \\
&det(J(\nabla M(X))) = M_{xx} M_{yy} - M_{xy} M_{yx} = (M_{xx})^2 + (M_{xy})^2 \geq 0
\end{aligned}
\tag{A.37}
$$

where C_i is a feature value (color) and R_i is radius of the i-th landmark at location ($X_i = (a_i, b_i)$) with the current location ($X = (x, y)$) and the home location ($X_0 = (x_0, y_0)$).

$det(J(\nabla M_1(X)))$ can be positive ($M_{xx} \neq 0$ or $M_{xy} \neq 0$) or zero ($M_{xx} = 0$ and $M_{xy} = 0$). Similarly, we obtain

$$\begin{aligned}
&\nabla M_2 = -\nabla M_1 = \sum_i (\tfrac{R_iC_i}{||X_i - X_0||^2})[-a_i + x_0, b_i - y_0] - (\tfrac{R_iC_i}{||X_i - X||^2})[-a_i + x, b_i - y] \\
&\nabla M_2(X_0) = 0, det(J(\nabla M_2(X))) \geq 0
\end{aligned} \tag{A.38}$$

where, $det(J(\nabla M_2(X)))$ can be positive ($M_{xx} \neq 0$ or $M_{xy} \neq 0$) or zero ($M_{xx} = 0$ and $M_{xy} = 0$).

$$\begin{aligned}
&\nabla M_3 = \sum_i (\tfrac{R_iC_i}{||X_i - X_0||^2})[b_i - y_0, a_i - x_0] - (\tfrac{R_iC_i}{||X_i - X||^2})[b_i - y, a_i - x] \\
&M_{xx} = \sum_i -2R_iC_i \tfrac{(x-a_i)(-y+b_i)}{||X_i - X||^4} = M_{yy} \\
&M_{xy} = \sum_i -R_iC_i \tfrac{(x-a_i)^2 - (y-b_i)^2}{||X_i - X||^4} = -M_{yx} \\
&det(J(\nabla M_3(X))) = M_{xx}M_{yy} - M_{xy}M_{yx} = (M_{xx})^2 + (M_{xy})^2 \geq 0
\end{aligned} \tag{A.39}$$

where, $det(J(\nabla M_3(X)))$ can be positive ($M_{xx} \neq 0$ or $M_{xy} \neq 0$) or zero ($M_{xx} = 0$ and $M_{xy} = 0$).

$$\begin{aligned}
&\nabla M_4 = -\nabla M_3 = \sum_i (\tfrac{R_iC_i}{||X_i - X||^2})[-b_i + y_0, -a_i + x_0] - (\tfrac{R_iC_i}{||X_i - X||^2})[-b_i + y, -a_i + x] \\
&\nabla M_4(X_0) = 0, det(J(\nabla M_4(X))) \geq 0
\end{aligned} \tag{A.40}$$

where, $det(J(\nabla M_4(X)))$ can be positive ($M_{xx} \neq 0$ or $M_{xy} \neq 0$) or zero ($M_{xx} = 0$ and $M_{xy} = 0$). All the cases can be written as

$$\begin{aligned}
&\nabla M_1(X_0) = 0, det(J(\nabla M_1(X))) \geq 0, (M_1)_{xx} = \sum_i -\eta R_iC_i \cos(2\theta_i)/D_i^2 = K_1 \\
&\nabla M_2(X_0) = 0, det(J(\nabla M_2(X))) \geq 0, (M_2)_{xx} = \sum_i \eta R_iC_i \cos(2\theta_i)/D_i^2 = -K_1 \\
&\nabla M_3(X_0) = 0, det(J(\nabla M_3(X))) \geq 0, (M_3)_{xx} = \sum_i -\eta R_iC_i \sin(2\theta_i)/D_i^2 = K_2 \\
&\nabla M_4(X_0) = 0, det(J(\nabla M_4(X))) \geq 0, (M_4)_{xx} = \sum_i \eta R_iC_i \sin(2\theta_i)/D_i^2 = -K_2
\end{aligned} \tag{A.41}$$

where we set K_1 and K_2 for simplicity. These four cases like $(M_k)_{xx}$ are quadratic differential about x in each case and the outputs can be represented by two constants

(K_1 and K_2). First, if these two constants are not zero, all the outputs can be written as

$$\left[\begin{array}{lll} K_1 > 0 & -K_1 < 0 & \nabla M_1 : success \\ K_1 < 0 & -K_1 > 0 & \nabla M_2 : success \\ K_2 > 0 & -K_2 < 0 & \nabla M_3 : success \\ K_2 < 0 & -K_2 > 0 & \nabla M_4 : success \end{array}\right. \tag{A.42}$$

For either $K_1 \neq 0$ or $K_2 \neq 0$, $det(J(\nabla M_k(X))) > 0$ since $(M_k)_{xx} > 0$. According to our analysis, there must be at most two or at least one model will have convergence. For example, the case with $K_1 = 0$ and $K_2 > 0$ has ∇M_3 as solution. If all of them are zero, $det(J(\nabla M_k)) = (M_{xx})^2 + (M_{xy})^2$ can also be zero. Through the above part, we check this condition and it has also homing property with special pattern. Attached to this property, we cannot know what type of model is right therefore we change models to find proper one in time-sequence.

Bibliography

Al-Naymat, G., Chawla, S., and Taheri, J. (2009). Sparsedtw: A novel approach to speed up dynamic time warping. In *Proceedings of the Eighth Australasian Data Mining Conference-Volume 101*, pages 117–127. Australian Computer Society, Inc.

Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). Freak: Fast retina keypoint. In *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*, pages 510–517. IEEE.

Anderson, A. (1977). A model for landmark learning in the honey-bee. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 114(3):335–355.

Angulo, C. and Godo, L. (2007). Using the average landmark vector method for robot homing. *Artificial intelligence research and development*, 163:331.

Aranda, M., Lopez-Nicolas, G., and Sagües, C. (2013). Sinusoidal input-based visual control for nonholonomic vehicles. *Robotica*, 31:811–823.

Aranda, M., Lopez-Nicolas, G., and Sagües, C. (2017). Angle-Based Navigation Using the 1D Trifocal Tensor. In *Control of Multiple Robots Using Vision Sensors*, pages 19–51. Springer.

Argyros, A., Bekris, K., Orphanoudakis, S., and Kavraki, L. (2005). Robot homing by exploiting panoramic vision. *Autonomous Robots*, 19(1):7–25.

Arroyo, R., Alcantarilla, P. F., Bergasa, L. M., Yebes, J. J., and Bronte, S. (2014). Fast and effective visual place recognition using binary codes and disparity information. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3089–3094. IEEE.

Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., and Baskurt, A. (2011). Sequential deep learning for human action recognition. In *Human Behavior Understanding*, pages 29–39. Springer.

Baddeley, B., Graham, P., Husbands, P., and Philippides, A. (2012). A model of ant route navigation driven by scene familiarity. *PLoS computational biology*, 8(1):e1002336.

Baddeley, B., Graham, P., Philippides, A., and Husbands, P. (2011). Holistic visual encoding of ant-like routes: Navigation without waypoints. *Adaptive Behavior*, 19(1):3–15.

Basten, K. and Mallot, H. (2010). Simulated visual homing in desert ant natural environments: efficiency of skyline cues. *Biological Cybernetics*, 102(5):413–425.

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359.

Berenguer, Y., Payá, L., Ballesta, M., and Reinoso, O. (2015). Position estimation and local mapping using omnidirectional images and global appearance descriptors. *Sensors*, 15(10):26368–26395.

Bouguet, J. (2001). Pyramidal implementation of the affine lucas kanade feature tracker—description of the algorithm. Technical report, Intel Corporation.

Buehlmann, C., Woodgate, J. L., and Collett, T. S. (2016). On the encoding of panoramic visual scenes in navigating wood ants. *Current Biology*, 26(15):2022–2027.

Burgard, W., Derr, A., Fox, D., and Cremers, A. (1998). Integrating global position estimation and position tracking for mobile robots: the dynamic markov localization approach. In *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, volume 2, pages 730–735. IEEE.

Calonder, M., Lepetit, V., Ozuysal, M., Trzcinski, T., Strecha, C., and Fua, P. (2012). Brief: Computing a local binary descriptor very fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1281–1298.

Cartwright, B. and Collett, T. (1983). Landmark learning in bees. *Journal of Comparative Physiology A*, 151(4):521–543.

Cartwright, B. and Collett, T. (1987). Landmark maps for honeybees. *Biological Cybernetics*, 57(1):85–93.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2016). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*.

Chen, S. Y. (2012). Kalman filter for robot vision: A survey. *IEEE Transactions on Industrial Electronics*, 59(11):4409–4420.

Cheung, A. and Vickerstaff, R. (2010). Finding the way with a noisy brain. *PLoS computational biology*, 6(11):e1000992.

Chung, W., Kim, H., Yoo, Y., Moon, C., and Park, J. (2012). The detection and following of human legs through inductive approaches for a mobile robot with a single laser range finder. *IEEE Transactions on Industrial Electronics*, 59(8):3156–3166.

Chung, W., Kim, S., Choi, M., Kim, H., Moon, C., and Song, J. (2009). Safe navigation of a mobile robot considering visibility of environment. *IEEE Transactions on Industrial Electronics*, 56(10):3941–3950.

Churchill, D. and Vardy, A. (2008). Homing in scale space. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008*, pages 1307–1312.

Churchill, D. and Vardy, A. (2013). An orientation invariant visual homing algorithm. *Journal of Intelligent & Robotic Systems*, pages 1–27.

Cireşan, D., Meier, U., Masci, J., and Schmidhuber, J. (2012). Multi-column deep neural network for traffic sign classification. *Neural Networks*, 32:333–338.

Collet, T. and Land, M. (1975). Visual spatial memory in a hoverfly. *Journal of Computational Physiology*, 100:59–84.

Collett, M. (2010). How desert ants use a visual landmark for guidance along a habitual route. *Proceedings of the National Academy of Sciences*, 107(25):11638.

Collett, M. (2012). How navigational guidance systems are combined in a desert ant. *Current Biology*.

Collett, M. and Collett, T. (2000). How do insects use path integration for their navigation? *Biological Cybernetics*, 83(3):245–259.

Collett, M., Graham, P., and Collett, T. S. (2017). Insect navigation: What backward walking reveals about the control of movement. *Current Biology*, 27(4):R141–R144.

Collett, T. (1996). Insect navigation en route to the goal: multiple strategies for the use of landmarks. *Journal of Experimental Biology*, 199(1):227–235.

Collett, T. S. and Graham, P. (2015). Insect navigation: do honeybees learn to follow highways? *Current Biology*, 25(6):R240–R242.

Collett, T. S., Wystrach, A., and Graham, P. (2016). Insect orientation: The travails of going straight. *Current Biology*, 26(11):R461–R463.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Conklin, J. and Eliasmith, C. (2005). A controlled attractor network model of path integration in the rat. *Journal of computational neuroscience*, 18(2):183–203.

Darwin, C. (1873). Origin of certain instincts. *Nature*, 7:417–418.

Davison, A. (2003). Real-time simultaneous localisation and mapping with a single camera. In *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings*, pages 1403–1410.

Davison, A., Cid, Y., and Kita, N. (2004). Real-time 3d slam with wide-angle vision. In *Proc IFAC Symposium on Intelligent Autonomous Vehicles Lisbon*. Citeseer.

Davison, A., Reid, I., Molton, N., and Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1052–1067.

Delgado, F. J., Quero, J. M., Garcia, J., Tarrida, C. L., Ortega, P. R., and Bermejo, S. (2012). Accurate and wide-field-of-view mems-based sun sensor for industrial applications. *IEEE Transactions on Industrial Electronics*, 59(12):4871–4880.

Dellaert, F., Fox, D., Burgard, W., and Thrun, S. (1999). Monte carlo localization for mobile robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1322–1328. IEEE.

Deng, H., Pan, C., Wen, T., and Liu, J. (2011). Entropy flow-aided navigation. *Journal of Navigation*, 64(1):109–125.

Deng, L., Hinton, G., and Kingsbury, B. (2013). New types of deep neural network learning for speech recognition and related applications: An overview. In *Acoustics,* *Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8599–8603. IEEE.

Denuelle, A., Strydom, R., and Srinivasan, M. V. (2015a). Snapshot-based control of uas hover in outdoor environments. In *Robotics and Biomimetics (ROBIO), 2015 IEEE International Conference on*, pages 1278–1284. IEEE.

Denuelle, A., Thurrowgood, S., Kendoul, F., and Srinivasan, M. V. (2015b). A view-based method for local homing of unmanned rotorcraft. In *Automation, Robotics and Applications (ICARA), 2015 6th International Conference on*, pages 443–449. IEEE.

Derdikman, D. and Knierim, J. J. (2014). *Space, time and memory in the hippocampal formation*. Springer.

Dissanayake, M., Newman, P., Clark, S., Durrant-Whyte, H., and Csorba, M. (2001). A solution to the simultaneous localization and map building (slam) problem. *Robotics and Automation, IEEE Transactions on*, 17(3):229–241.

Dittmar, L., Stürzl, W., Baird, E., Boeddeker, N., and Egelhaaf, M. (2010). Goal seeking in honeybees: matching of optic flow snapshots? *Journal of Experimental Biology*, 213(17):2913–2923.

Dong, J., Fei, X., Karianakis, N., Tsotsos, K., and Soatto, S. (2016). Vl-slam: Real-time visual-inertial navigation and semantic mapping. In *IEEE Conference on Computer Vision and Pattern Recognition, Live Demo*.

Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part i. *Robotics & Automation Magazine, IEEE*, 13(2):99–110.

El Jundi, B., Foster, J. J., Khaldy, L., Byrne, M. J., Dacke, M., and Baird, E. (2016). A snapshot-based mechanism for celestial orientation. *Current Biology*, 26(11):1456–1462.

el Jundi, B. and Homberg, U. (2010). Evidence for the possible existence of a second polarization-vision pathway in the locust brain. *Journal of Insect Physiology*, 56(8):971–979.

Esch, H., Zhang, S., Srinivasan, M., and Tautz, J. (2001). Honeybee dances communicate distances measured by optic flow. *Nature*, 411(6837):581–583.

Ess, A., Schindler, K., Leibe, B., and Van Gool, L. (2010). Object detection and tracking for autonomous navigation in dynamic environments. *The International Journal of Robotics Research*, 29(14):1707–1725.

Etienne, A. and Jeffery, K. (2004). Path integration in mammals. *Hippocampus*, 14(2):180–192.

Etienne, A., Maurer, R., and Sëguinot, V. (1996). Path integration in mammals and its interaction with visual landmarks. *Journal of Experimental Biology*, 199:201–209.

Eustice, R., Walter, M., and Leonard, J. (2005). Sparse extended information filters: Insights into sparsification. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3281–3288. IEEE.

Exner, D., Bruns, E., Kurz, D., Grundhöfer, A., and Bimber, O. (2010). Fast and robust camshift tracking. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 9–16. IEEE.

Fan, C., Chen, Z., Jacobson, A., Hu, X., and Milford, M. (2017). Biologically-inspired visual place recognition with adaptive multiple scales. *Robotics and Autonomous Systems*.

Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. *Image analysis*, pages 363–370.

Fiala, M. and Basu, A. (2004). Robot navigation using panoramic tracking. *Pattern Recognition*, 37(11):2195–2215.

Fleer, D. and Möller, R. (2017). Comparing holistic and feature-based visual methods for estimating the relative pose of mobile robots. *Robotics and Autonomous Systems*, 89:51–74.

Fox, D. (1998). *Markov localization: A probabilistic framework for mobile robot localization and navigation*. PhD thesis, PhD thesis, Dept. of Computer Science, University of Bonn, Germany.

Fox, D., Burgard, W., and Thrun, S. (1998). Active markov localization for mobile robots. *Robotics and Autonomous Systems*, 25(3):195–207.

Franz, M. (1999). *Minimalistic visual navigation=, Minimalistische visuelle Navigation*. PhD thesis, Universitat Tubingen.

Franz, M., Schölkopf, B., Mallot, H., and Bülthoff, H. (1998). Where did I take that snapshot? Scene-based homing by image matching. *Biological Cybernetics*, 79(3):191–202.

Frost, B. and Mouritsen, H. (2006). The neural mechanisms of long distance animal navigation. *Current opinion in neurobiology*, 16(4):481–488.

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., and Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914.

Gaffin, D. D. and Brayfield, B. P. (2016). Autonomous visual navigation of an indoor environment using a parsimonious, insect inspired familiarity algorithm. *PloS one*, 11(4):e0153706.

Garm, A. and Nilsson, D.-E. (2014). Visual navigation in starfish: first evidence for the use of vision and eyes in starfish. 281(1777):20133011.

Garm, A., Oskarsson, M., and Nilsson, D. (2011). Box jellyfish use terrestrial visual cues for navigation. *Current Biology*.

Gerstmayr, L., Röben, F., Krzykawski, M., Kreft, S., Venjakob, D., and Möller, R. (2009). A vision-based trajectory controller for autonomous cleaning robots. *Autonome Mobile Systeme 2009*, pages 65–72.

Giachetti, A., Campani, M., and Torre, V. (1998). The use of optical flow for road navigation. *Robotics and Automation, IEEE Transactions on*, 14(1):34–48.

Gilg, A. and Schmidt, G. (1994a). Landmar-oriented visual navigation of a mobile robot. *IEEE Transactions on Industrial Electronics*, 41(4):392–397.

Gilg, A. and Schmidt, G. (1994b). Landmark-oriented visual navigation of a mobile robot. *IEEE Transactions on Industrial Electronics*, 41(4):392–397.

Goedemé, T., Tuytelaars, T., Van Gool, L., Vanacker, G., and Nuttin, M. (2005). Feature based omnidirectional sparse visual path following. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005.(IROS 2005)*, pages 1806–1811.

Goldhoorn, A., Ramisa, A., de Mantaras, R., and Toledo, R. (2007). Using the average landmark vector method for robot homing. *Frontiers in artificial intelligence and applications*, 163:331–338.

Gould, J. L. (2015). Animal navigation: Birds have magnetic maps. *Current Biology*, 25(19):R836–R838.

Graham, P. and Philippides, A. (2017). Vision for navigation: What can we learn from ants? *Arthropod Structure & Development*.

Graham, P., Philippides, A., and Baddeley, B. (2010). Animal Cognition: Multi-modal Interactions in Ant Learning. *Current Biology*, 20(15):R639–R640.

Gupta, M., Arunkumar, G., and Vachhani, L. (2017a). Bearing only visual homing: Observer based approach. In *Control and Automation (MED), 2017 25th Mediterranean Conference on*, pages 358–363. IEEE.

Gupta, S., Davidson, J., Levine, S., Sukthankar, R., and Malik, J. (2017b). Cognitive mapping and planning for visual navigation. *arXiv preprint arXiv:1702.03920*.

Haferlach, T., Wessnitzer, J., Mangan, M., and Webb, B. (2007). Evolving a neural model of insect path integration. *Adaptive Behavior*, 15(3):273–287.

Hafner, V. (2001). Adaptive Homing–Robotic Exploration Tours. *Adaptive Behavior*, 9(3-4):131–141.

Hahnel, D., Burgard, W., Fox, D., and Thrun, S. (2003). An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 206–211. IEEE.

Hamaoui, M. (2017). Polarized skylight navigation. *Applied Optics*, 56(3):B37–B46.

Hong, J., Tan, X., Pinette, B., Weiss, R., and Riseman, E. (1992). Image-based homing. *Control Systems Magazine, IEEE*, 12(1):38–45.

Horst, M. and Möller, R. (2017). Visual place recognition for autonomous mobile robots. *Robotics*, 6(2):9.

Huang, X. and Yan, Y. (2016). Output feedback control of underactuated spacecraft hovering in circular orbit with radial or in-track controller failure. *IEEE Transactions on Industrial Electronics*, 63(9):5569–5581.

Huber, E. and Kortenkamp, D. (2002). Using stereo vision to pursue moving agents with a mobile robot. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 3, pages 2340–2346. IEEE.

Hübner, W. (2005). *From homing behavior to cognitive mapping: Integration of egocentric pose relations and allocentric landmark information in a graph model*. PhD thesis, Ph. D. thesis, Fachbereich 3 (Mathematik & Informatik), Universität Bremen.

Hwang, S. and Song, J. (2011). Monocular vision-based slam in indoor environment using corner, lamp, and door features from upward-looking camera. *IEEE Transactions on Industrial Electronics*, 58(10):4804–4812.

Issa, J. and Zhang, K. (2012). Universal conditions for exact path integration in neural systems. *Proceedings of the National Academy of Sciences*, 109(17):6716–6720.

Jensen, C. S., Lu, H., and Yang, B. (2009). Graph model based indoor tracking. In *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*, pages 122–131. IEEE.

Jouir, T., Strydom, R., and Srinivasan, M. V. (2015). A 3d sky compass to achieve robust estimation of uav attitude. In *Australasian Conference on Robotics and Automation (ACRA)*. Australian Robotics and Automation Association.

Kang, Y., Kim, H., Ryu, S., Doh, N., Oh, Y., and You, B. (2012). Dependable humanoid navigation system based on bipedal locomotion. *IEEE Transactions on Industrial Electronics*, 59(2):1050–1060.

Karmaker, D., Schiffner, I., Strydom, R., and Srinivasan, M. V. (2016). Whog: A weighted hog-based scheme for the detection of birds and identification of their poses in natural environments. In *Control, Automation, Robotics and Vision (ICARCV), 2016 14th International Conference on*, pages 1–7. IEEE.

Ke, Y. and Sukthankar, R. (2004). Pca-sift: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE.

Kim, D. E. and Lee, J. (2011). Path integration mechanism with coarse coding of neurons. *Neural processing letters*, pages 1–15.

Kim, Y.-K., Kim, Y., Y., J., Jang, G., Kim, K.-S., Kim, S., and Kwak, B. (2013). Developing accurate long-distance 6-dof motion detection with one-dimensional laser sensors: Three-beam detection system. *IEEE Transactions on Industrial Electronics*, 60(8):3386–3395.

Kimchi, T., Etienne, A., and Terkel, J. (2004). A subterranean mammal uses the magnetic compass for path integration. *Proceedings of the National Academy of Sciences of the United States of America*, 101(4):1105.

Kirchner, W. and Braun, U. (1994). Dancing honey bees indicate the location of food sources using path integration rather than cognitive maps. *Animal behaviour*, 48(6):1437–1441.

Kohler, M. and Wehner, R. (2005). Idiosyncratic route-based memories in desert ants, melophorus bagoti: How do they interact with path-integration vectors? *Neurobiology of Learning and Memory*, 83(1):1–12.

Kubie, J. and Fenton, A. (2009). Heading-vector navigation based on head-direction cells and path integration. *Hippocampus*, 19(5):456–479.

Kuglin, C. D. and Hines, D. C. (1975). The phase correlation image alignment method. In *Proceedings International Conference on Cybernetics and Society*, pages 163–165.

Kuipers, B. and Byun, Y. (1988). Robust, qualitative approach to a spatial learning mobile robot. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 1003, page 366.

Kuipers, B. and Byun, Y. (1991). A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and autonomous systems*, 8(1-2):47–63.

Kwon, T. and Song, J. (2011a). A new feature commonly observed from air and ground for outdoor localization with elevation map built by aerial mapping system. *Journal of Field Robotics*, 28(2):227–240.

Kwon, T. and Song, J. (2011b). A new feature commonly observed from air and ground for outdoor localization with elevation map built by aerial mapping system. *Journal of Field Robotics*, 28(2):227–240.

Kwon, T., Song, J., and Joo, S. (2010). Elevation moment of inertia: A new feature for monte carlo localization in ourdoor environment with elevation map. *Journal of Field Robotics*, 27(3):371–386.

Labrosse, F. (2006). The visual compass: performance and limitations of an appearance-based method. *Journal of Field Robotics*, 23(10):913–941.

Labrosse, F. (2007). Short and long-range visual navigation using warped panoramic images. *Robotics and Autonomous Systems*, 55(9):675–684.

Lambrinos, D., Moller, R., Labhart, T., Pfeifer, R., and Wehner, R. (2000). A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, 30(1-2):39–64.

Lee, C. and Kim, D. (2017a). Local homing navigation based on the moment model for landmark distribution and features. *Sensors*, 17(11):2658.

Lee, C. and Kim, D. E. (2015). Image reconstruction-based solution for occlusion problems in visual navigation systems. In *ITC-CSCC: International Technical Conference on Circuits Systems, Computers and Communications*, pages 878–879.

Lee, C. and Kim, D. E. (2016a). A landmark vector approach using gray-colored information. In *International Conference on Simulation of Adaptive Behavior*, pages 138–144. Springer.

Lee, C. and Kim, D. E. (2016b). A moment measure model of landmarks for local homing navigation. In *International Conference on Simulation of Adaptive Behavior*, pages 126–137. Springer.

Lee, C. and Kim, D. E. (2017b). Analysing the homing navigation using various moment models based on landmark characteristics (being prepared). *IEEE transactions on cybernetics*.

Lee, C. and Kim, D. E. (2017c). Dynamic time warping approach to visual homing navigation (being prepared). *IEEE transactions on cybernetics*.

Lee, C. and Kim, D. E. (2017d). Visual homing navigation based on bipolar-like features (being prepared). *Bioinspiration & Biomimetics*.

Lee, C. and Kim, D. E. (2017e). Visual homing navigation using haar-like features and their matching (under review). *Sensors*.

Lee, C. and Kim, D. E. (2017f). Visual navigation based on pixelwise landmark information (being prepared). *Sensors*.

Lee, C., Yu, S.-E., and Kim, D. E. (2017). Landmark-based homing navigation using omnidirectional depth information. *Sensors*, 17(8):1928.

Lee, D., Kim, G., Kim, D., Myung, H., and Choi, H.-T. (2012). Vision-based object detection and tracking for autonomous navigation of underwater robots. *Ocean Engineering*, 48:59–68.

Legge, E., Spetch, M., and Cheng, K. (2010). Not using the obvious: desert ants, melophorus bagoti, learn local vectors but not beacons in an arena. *Animal cognition*, 13(6):849–860.

Lent, D., Graham, P., and Collett, T. (2010). Image-matching during ant navigation occurs through saccade-like body turns controlled by learned visual features. *Proceedings of the National Academy of Sciences*, 107(37):16348–16353.

Lent, D. D., Graham, P., and Collett, T. S. (2013). Phase-dependent visual control of the zigzag paths of navigating wood ants. *Current Biology*, 23(23):2393–2399.

Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). Brisk: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555. IEEE.

Lienhart, R. and Maydt, J. (2002). An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–900. IEEE.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2016). Feature pyramid networks for object detection. *arXiv preprint arXiv:1612.03144*.

Liu, M., Pradalier, C., and Siegwart, R. (2013). Visual homing from scale with an uncalibrated omnidirectional camera. *IEEE Transactions on Robotics*, 29(6):1353–1365.

Lohmann, K. J., Lohmann, C. M., Ehrhart, L. M., Bagley, D. A., and Swing, T. (2004). Animal behaviour: geomagnetic map used in sea-turtle navigation. *Nature*, 428(6986):909–910.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

Lucas, B. D., Kanade, T., et al. (1981). An iterative image registration technique with an application to stereo vision.

Luke, R. H., Keller, J. M., Skubic, M., and Senger, S. (2005). Acquiring and maintaining abstract landmark chunks for cognitive robot navigation. In *Intelligent Robots* *and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 2566–2571. IEEE.

Luschi, P., Papi, F., Liew, H., Chan, E., and Bonadonna, F. (1996). Long-distance migration and homing after displacement in the green turtle (Chelonia mydas): a satellite tracking study. *Journal of Comparative Physiology A*, 178(4):447–452.

Lyshevski, S. E. (2017). Signal processing in cyber-physical mems sensors: Inertial measurement and navigation systems. *IEEE Transactions on Industrial Electronics*.

Ma, J., Zhao, J., Guo, H., Jiang, J., Zhou, H., and Gao, Y. (2017). Locality preserving matching.

Mangan, M. and Webb, B. (2009). Modelling place memory in crickets. *Biological cybernetics*, 101(4):307–323.

Mather, J. (1991). Navigation by spatial memory and use of visual landmarks in octopuses. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 168(4):491–497.

McNaughton, B., Battaglia, F., Jensen, O., Moser, E., and Moser, M. (2006). Path integration and the neural basis of the'cognitive map'. *Nature Reviews Neuroscience*, 7(8):663–678.

Méndez-Polanco, J., Muñoz Meléndez, A., and Morales, E. (2009). People Detection by a Mobile Robot Using Stereo Vision in Dynamic Indoor Environments. *MICAI 2009: Advances in Artificial Intelligence*, pages 349–359.

Merkle, T. and Wehner, R. (2010). Desert ants use foraging distance to adapt the nest search to the uncertainty of the path integrator. *Behavioral Ecology*, 21(2):349–355.

Meyer, J. and Filliat, D. (2003). Map-based navigation in mobile robots:: Ii. a review of map-learning and path-planning strategies. *Cognitive Systems Research*, 4(4):283–317.

Mita, T., Kaneko, T., and Hori, O. (2005). Joint haar-like features for face detection. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1619–1626. IEEE.

Mittelstaedt, M. and Mittelstaedt, H. (1980). Homing by path integration in a mammal. *Naturwissenschaften*, 67(11):566–567.

Möller, R. (2000). Insect visual homing strategies in a robot with analog processing. *Biological Cybernetics*, 83(3):231–243.

Möller, R. (2009). Local visual homing by warping of two-dimensional images. *Robotics and Autonomous Systems*, 57(1):87–101.

Möller, R. (2012). A model of ant navigation based on visual prediction. *Journal of theoretical biology*, 305:118–130.

Möller, R. (2016a). Column distance measures and their effect on illumination tolerance in minwarping. Technical report, Tech. rep., University of Bielefeld, Faculty of Technology, Computer Engineering Group, 2016 http://www. ti. uni-bielefeld. de/html/people/moeller/tsimd_warpingsimd. html.

Möller, R. (2016b). A simd implementation of the minwarping method for local visual homing. Technical report, Technical report, Computer Engineering, Faculty of Technology, Bielefeld University.

Möller, R., Horst, M., and Fleer, D. (2014). Illumination tolerance for visual navigation with the holistic min-warping method. *Robotics*, 3(1):22–67.

Möller, R., Krzykawski, M., and Gerstmayr, L. (2010). Three 2d-warping schemes for visual robot navigation. *Autonomous Robots*, 29(3):253–291.

Möller, R., Maris, M., and Lambrinos, D. (1999). A neural model of landmark navigation in insects. *Neurocomputing*, 26:801–808.

Möller, R. and Vardy, A. (2006). Local visual homing by matched-filter descent in image distances. *Biological Cybernetics*, 95(5):413–430.

Möller, R., Vardy, A., Kreft, S., and Ruwisch, S. (2007). Visual homing in environments with anisotropic landmark distribution. *Autonomous Robots*, 23(3):231–245.

Montemerlo, M. and Thrun, S. (2007). Fastslam 2.0. *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*, pages 63–90.

Montiel, J. and Davison, A. (2006). A visual compass based on slam. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1917–1922. IEEE.

Morel, J.-M. and Yu, G. (2009). Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469.

Moser, E. I., Kropff, E., and Moser, M.-B. (2008). Place cells, grid cells, and the brain's spatial representation system. *Annual review of neuroscience*, 31.

Müller, M., Mattes, H., and Kurth, F. (2006). An efficient multiscale approach to audio synchronization. In *ISMIR*, pages 192–197.

Murillo, A. C., Guerrero, J. J., and Sagues, C. (2007). Surf features for efficient robot localization with omnidirectional images. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3901–3907. IEEE.

Narendra, A. (2007). Homing strategies of the australian desert ant melophorus bagoti ii. interaction of the path integrator with visual cue information. *Journal of Experimental Biology*, 210(10):1804–1812.

Nelson, R. and Aloimonos, J. (1988). Finding motion parameters from spherical motion fields (or the advantages of having eyes in the back of your head). *Biological Cybernetics*, 58(4):261–273.

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. (2011). Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696.

Nouar, O.-D., Ali, G., and Raphael, C. (2006). Improved object tracking with camshift algorithm. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 2, pages II–II. IEEE.

Nummiaro, K., Koller-Meier, E., and Van Gool, L. (2003). An adaptive color-based particle filter. *Image and Vision Computing*, 21(1):99–110.

O'Keefe, J. and Burgess, N. (1996). Geometric determinants of the place fields of hippocampal neurons. *Nature*, 381(6581):425–428.

Ouyang, W. and Wang, X. (2013). Joint deep learning for pedestrian detection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2056–2063. IEEE.

Paik, S.-B. and Ringach, D. L. (2011). Retinal origin of orientation maps in visual cortex. *Nature neuroscience*, 14(7):919–925.

Pan, C., Deng, H., Yin, X., and Liu, J. (2011). An optical flow-based integrated navigation system inspired by insect vision. *Biological cybernetics*, pages 1–14.

Papi, F. (1990). Olfactory navigation in birds. *Cellular and Molecular Life Sciences*, 46(4):352–363.

Park, S. and Hashimoto, S. (2009). Autonomous mobile robot navigation using passive rfid in indoor environment. *IEEE Transactions on Industrial Electronics*, 56(7):2366–2373.

Pons, J., Hübner, W., Dahmen, H., and Mallot, H. (2007). Vision-based robot homing in dynamic environments. In *13th IASTED International Conference on Robotics and Applications, K. Schilling, Ed*, pages 293–298.

Ramisa, A., Goldhoorn, A., Aldavert, D., Toledo, R., and de Mantaras, R. (2011). Combining invariant features and the alv homing method for autonomous robot navigation based on panoramas. *Journal of Intelligent & Robotic Systems*, pages 1–25.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788.

Reid, S., Narendra, A., Hemmi, J., and Zeil, J. (2011). Polarised skylight and the landmark panorama provide night-active bull ants with compass information during route following. *Journal of Experimental Biology*, 214(3):363–370.

Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149.

Reppert, S., Gegear, R., and Merlin, C. (2010). Navigational mechanisms of migrating monarch butterflies. *Trends in neurosciences*, 33(9):399–406.

Rossier, J., Haeberli, C., and Schenk, F. (2000). Auditory cues support place navigation in rats when associated with a visual cue. *Behavioural Brain Research*, 117(1-2):209–214.

Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. *Computer Vision–ECCV 2006*, pages 430–443.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE.

Ruck, D. W., Rogers, S. K., Kabrisky, M., Oxley, M. E., and Suter, B. W. (1990). The multilayer perceptron as an approximation to a bayes optimal discriminant function. *Neural Networks, IEEE Transactions on*, 1(4):296–298.

Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49.

Samsonovich, A. and McNaughton, B. (1997). Path integration and cognitive mapping in a continuous attractor neural network model. *The Journal of Neuroscience*, 17(15):5900–5920.

Sasaki, T., Brscic, D., and Hashimotor, H. (2010). Human-observation-based extraction of path patterns for mobile robot navigation. *IEEE Transactions on Industrial Electronics*, 57(4):1401–1410.

Schwarz, S. and Cheng, K. (2010). Visual associative learning in two desert ant species. *Behavioral Ecology and Sociobiology*, 64(12):2033–2041.

Schwarz, S., Mangan, M., Zeil, J., Webb, B., and Wystrach, A. (2017). How ants use vision when homing backward. *Current Biology*, 27(3):401–407.

Schwarz, S., Narendra, A., and Zeil, J. (2011). The properties of the visual system in the australian desert ant melophorus bagoti. *Arthropod Structure & Development*, 40(2):128–134.

Silva, D. F. and Batista, G. E. (2016). Speeding up all-pairwise dynamic time warping matrix calculation. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 837–845. SIAM.

Smith, L., Philippides, A., Graham, P., Baddeley, B., and Husbands, P. (2007). Linked local navigation for visual route guidance. *Adaptive Behavior*, 15(3):257–271.

Srinivasan, M. (2010). Honey bees as a model for vision, perception, and cognition. *Annual review of entomology*, 55:267–284.

Srinivasan, M. V. (2017). How lost "passenger" ants find their way home. *Learning & Behavior*, pages 1–2.

Steck, K., Knaden, M., and Hansson, B. (2010). Do desert ants smell the scenery in stereo? *Animal Behaviour*, 79(4):939–945.

Strydom, R., Denuelle, A., and Srinivasan, M. V. (2016). Bio-inspired principles applied to the guidance, navigation and control of uas. *Aerospace*, 3(3):21.

Stürzl, W., Grixa, I., Mair, E., Narendra, A., and Zeil, J. (2015). Three-dimensional models of natural environments and the mapping of navigational information. *Journal of Comparative Physiology A*, 201(6):563–584.

Stürzl, W. and Mallot, H. (2006). Efficient visual homing based on fourier transformed panoramic images. *Robotics and Autonomous Systems*, 54(4):300–313.

Stürzl, W. and Möller, R. (2007). An insect-inspired active vision approach for orientation estimation with panoramic images. *Bio-inspired Modeling of Cognitive Tasks*, pages 61–70.

Stürzl, W. and Zeil, J. (2007). Depth, contrast and view-based homing in outdoor scenes. *Biological Cybernetics*, 96(5):519–531.

Stürzl, W., Zeil, J., Boeddeker, N., and Hemmi, J. M. (2016). How wasps acquire and use views for homing. *Current Biology*, 26(4):470–482.

Sun, Y., Wang, X., and Tang, X. (2014). Deep learning face representation from predicting 10,000 classes. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1891–1898. IEEE.

Szegedy, C., Erhan, D., and Toshev, A. T. (2016). Object detection using deep neural networks. US Patent 9,275,308.

Thrun, S. and Bücken, A. (1996). Integrating grid-based and topological maps for mobile robot navigation. In *Proceedings of the National Conference on Artificial Intelligence*, pages 944–951.

Thrun, S., Fox, D., Burgard, W., and Dellaert, F. (2001). Robust monte carlo localization for mobile robots. *Artificial intelligence*, 128(1):99–141.

Thrun, S. and Montemerlo, M. (2006). The graph slam algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, 25(5-6):403–429.

Trullier, O., Meyer, J., et al. (1997). Biomimetic navigation models and strategies in animats. *AI communications*, 10(2):79–92.

Ugolini, A., Borgioli, G., Galanti, G., Mercatelli, L., and Hariyama, T. (2010). Photoresponses of the compound eye of the sandhopper talitrus saltator (crustacea, amphipoda) in the ultraviolet-blue range. *The Biological Bulletin*, 219(1):72–79.

Vardy, A. (2006). Long-range visual homing. In *Robotics and Biomimetics, 2006. ROBIO'06. IEEE International Conference on*, pages 220–226. IEEE.

Vardy, A. and Möller, R. (2005). Biologically plausible visual homing methods based on optical flow techniques. *Connection Science*, 17(1):47–89.

Vickerstaff, R. and Cheung, A. (2010). Which coordinate system for modelling path integration? *Journal of theoretical biology*, 263(2):242–261.

Vickerstaff, R. and Merkle, T. (2012). Path integration mediated systematic search: a bayesian model. *Journal of Theoretical Biology*.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE.

Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57(2):137–154.

Wajnberg, E., Acosta-Avalos, D., Alves, O., de Oliveira, J., Srygley, R., and Esquivel, D. (2010). Magnetoreception in eusocial insects: an update. *Journal of The Royal Society Interface*, 7(Suppl 2):S207–S225.

Walker, C., Graham, P., and Philippides, A. (2017). Using deep autoencoders to investigate image matching in visual navigation. In *Conference on Biomimetic and Biohybrid Systems*, pages 465–474. Springer.

Wang, W. and Xie, G. (2015). Online high-precision probabilistic localization of robotic fish using visual and inertial cues. *IEEE Transactions on Industrial Electronics*, 62(2):1113–1124.

Wang, Z., Yang, X., Xu, Y., and Yu, S. (2009). Camshift guided particle filter for visual tracking. *Pattern Recognition Letters*, 30(4):407–413.

Weber, K., Venkatesh, S., and Srinivasan, M. (1999). Insect-inspired robotic homing. *Adaptive Behavior*, 7(1):65–97.

Wehner, R. (2016). Early ant trajectories: spatial behaviour before behaviourism. *Journal of Comparative Physiology A*, 202(4):247–266.

Wehner, R., Michel, B., and Antonsen, P. (1996a). Visual navigation in insects: coupling of egocentric and geocentric information. *Journal of Experimental Biology*, 199:129–140.

Wehner, R., Michel, B., and Antonsen, P. (1996b). Visual navigation in insects: coupling of egocentric and geocentric information. *Journal of Experimental Biology*, 199:129–140.

Wehner, R. and Räber, F. (1979). Visual spatial memory in desert ants, Cataglyphis bicolor (Hymenoptera: Formicidae). *Cellular and Molecular Life Sciences*, 35(12):1569–1571.

Wehner, R. and Wehner, S. (1990). Insect navigation: use of maps or ariadne's thread? *Ethology Ecology & Evolution*, 2(1):27–48.

Widyotriatmo, A. and Hong, K.-S. (2011). Navigation function-based control of multiple wheeled vehicles. *IEEE Transactions on Industrial Electronics*, 58(5):1896–1906.

Willis, M. (2008). Chemical plume tracking behavior in animals and mobile robots. *Navigation*, 55(2):127.

Wintergerst, S. and Ronacher, B. (2012). Discrimination of inclined path segments by the desert ant cataglyphis fortis. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, pages 1–11.

Wittmann, T. and Schwegler, H. (1995). Path integration—a network model. *Biological Cybernetics*, 73(6):569–575.

Wolf, J., Burgard, W., and Burkhardt, H. (2005). Robust vision-based localization by combining an image-retrieval system with monte carlo localization. *Robotics, IEEE Transactions on*, 21(2):208–216.

Wu, H., Wu, Y.-X., Liu, C.-A., Yang, G.-T., and Qin, S.-Y. (2016). Fast robot localization approach based on manifold regularization with sparse area features. *Cognitive Computation*, 8(5):856–876.

Wystrach, A., Beugnon, G., and Cheng, K. (2012). Ants might use different view-matching strategies on and off the route. *The Journal of Experimental Biology*, 215(1):44–55.

Wystrach, A., Dewar, A., Philippides, A., and Graham, P. (2016). How do field of view and resolution affect the information content of panoramic scenes for visual navigation? a computational investigation. *Journal of Comparative Physiology A*, 202(2):87–95.

Wystrach, A., Schwarz, S., Schultheiss, P., Beugnon, G., and Cheng, K. (2011). Views, landmarks, and routes: how do desert ants negotiate an obstacle course? *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 197(2):167–179.

Yang, X. and Cheng, K.-T. (2012). Ldb: An ultra-fast feature for scalable augmented reality on mobile devices. In *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, pages 49–57. IEEE.

Yu, S.-E. and Kim, D. E. (2010a). Discretized landmark distance estimation method for mobile robot navigation. *Korea Robotics Society Annual Conference 2010*, pages 136–137.

Yu, S.-E. and Kim, D. E. (2010b). Distance estimation method with snapshot landmark images in the robotic homing navigation. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 275–280. IEEE.

Yu, S.-E. and Kim, D. E. (2011a). Image-based homing navigation with landmark arrangement matching. *Information sciences*, 181(16):3427–3442.

Yu, S.-E. and Kim, D. E. (2011b). Landmark vectors with quantized distance information for homing navigation. *Adaptive Behavior*, 19(2):121–141.

Yu, S.-E. and Kim, D. E. (2012). Analyzing the effect of landmark vectors in homing navigation. *Adaptive Behavior*, 20(5):337–359.

Zeil, J. (2012). Visual homing: an insect perspective. *Current Opinion in Neurobiology*.

Zeil, J. and Hemmi, J. (2006). The visual ecology of fiddler crabs. *Journal of Comparative Physiology A*, 192(1):1–25.

Zeil, J., Hofmann, M., and Chahl, J. (2003). Catchment areas of panoramic snapshots in outdoor scenes. *Journal of the Optical society of America A*, 20(3):450–469.

Zeil, J. and Layne, J. (2002). Path integration in fiddler crabs and its relation to habitat and social life. *Crustacean experimental systems in neurobiology*, pages 227–246.

Zeil, J., Narendra, A., and Stürzl, W. (2014). Looking and homing: how displaced ants decide where to go. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 369(1636):20130034.

Zhang, T., Huang, S., Liu, D., Shi, L., Zhou, C., and Xiong, R. (2016). A method of state estimation for underwater vehicle navigation around a cylindrical structure. In *Industrial Electronics and Applications (ICIEA), 2016 IEEE 11th Conference on*, pages 101–106. IEEE.

Zhu, B., Zaini, A. H. B., and Xie, L. (2017a). Distributed guidance for interception by using multiple rotary-wing unmanned aerial vehicles. *IEEE Transactions on Industrial Electronics*.

Zhu, Q., Liu, C., and Cai, C. (2014). A Robot Navigation Algorithm Based on Sparse Landmarks. In *Proceedings of the 6th Int. Conf. on Intelligent Human-Machine Systems and Cybernetics*. IEEE.

Zhu, Q., Liu, C., and Cai, C. (2015). A novel robot visual homing method based on sift features. *Sensors*, 15(10):26063–26084.

Zhu, Y., Mottaghi, R., Kolve, E., Lim, J. J., Gupta, A., Fei-Fei, L., and Farhadi, A. (2017b). Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 3357–3364. IEEE.

Zollikofer, C., Wehner, R., and Fukushi, T. (1995). Optical scaling in conspecific cataglyphis ants. *Journal of Experimental Biology*, 198(8):1637–1646.