Visual Homing Navigation Based on Optical Flow

Youngseo Cha

The Graduate School
School of Electrical and Electronic Engineering
Yonsei University

Visual Homing Navigation Based on Optical Flow

A Masters Thesis
Submitted to the Department of
Electrical and Electronic Engineering
and the Graduate School of Yonsei University
in partial fulfillment of the
requirements for the degree of
Master of Engineering

Youngseo Cha

December 2012

This certifies that the master's thesis
of Youngseo Cha is approved.

Thesis Supervisor: DaeEun Kim

Sanghoon Lee

Euntai Kim

The Graduate School
Yonsei University
December 2012

Abstract

Visual navigation is one of the popular fields for robotic navigation. Generally, several problems limit the use of agent navigation with a vision sensor. For instance, hidden landmarks in a scene can degrade the performance of an estimating agent's state by distorting the visual descriptor. A high computational load can be also regarded as a critical problem originating from image processing with the high dimensional information of a measurement model. In this paper, we propose a bio-inspired model that can compensate for the defects of visual navigation, such as occlusion of landmarks, computational load, etc. The proposed model introduces a new probabilistic localization that recognizes the agent's state without preliminary exploration by using the desired map and an omni-directional image taken from a reference position.

Essentially, the proposed method assumes a snapshot hypothesis, which is one of the important features of a bio-inspired navigation model. With the snapshot hypothesis, the agent obtains two images, one taken from a reference position and the other from its current position. The differences in these images are then important factors for estimating the state of the agent from a reference position. This idea arises from monotonous changes that occur in the image as the agent moves a short distance from the reference position to another specific position. For example, according to the average landmark vector (ALV) method, which finds a homing vector from the current position by calculating the deviations in snapshot images, the spatial state of an agent can be estimated easily using two images. Among the many algorithms used to calculate and simplify differences between images, we use the Kanade-Lucas-Tomasi (KLT) algorithm in this paper to measure the deviation of two images to estimate the state of our agent. The KLT algorithm is a popular algorithm for optical flow that demonstrates differences between images with low computational load and selects the corners of the image automatically without any pre-learning phases.

Although a bio-inspired navigation model has the advantage of a fast processing speed, previous research has not used probabilistic approaches to study filtering of noise in the environment. One of the main reasons is the computational inefficiency of the probabilistic model. The proposed model solves this problem by using only a one-dimensional measurement model to apply a Bayesian filter based on the desired map. Even though this reduces the amount of information, the specification of a measurement model that has centralized homing vectors made from a snapshot hypothesis can help to localize the agent easily due to the dependence between the agent position and

the homing vector. The proposed model with probabilistic approaches makes two new contributions when compared with existing algorithms. First, no pre-searching is required to store measurement patterns of positions on the map. Instead, only one omnidirectional image, taken from the reference position, is sufficient for localizing the agent to estimate spatial information. Second, the proposed model can be performed nearly in real time with low level computation despite the use of probabilistic processing.

Acknowledgements

석사 과정 2년은 저에게 많은 고민을 안겨 주었던 시기였습니다. 만약 먼 미래에 이 시기를 돌이켜보게 된다면 제 인생에서 꽤 유별났던 기간이었다고 이야기할 수 있을 것 같습니다. 대학 내내 부단히도 저를 괴롭히던 생활들과 잠시 이별하고, 짧으나마 연구에 집중할 수 있었던 시간들은 저에게 있어 분에 넘치는 여유이자, 크나큰 책임이기도 했습니다. 이제 저 자신에게 주었던 유예를 마치고 애증의 생활들로 돌아가게 되었습니다. 이 졸업 논문은 그 기간 동안의 저를 설명하는 작은 증거품이 아닐까 생각합니다.

지도 교수님이신 김대은 교수님을 비롯하여 제 논문을 심사해주신 이상훈 교수님, 그리고 김은태 교수님께 감사드립니다. 연구를 진행하는 대학원생에게 있어, 연구 내적인 측면 뿐만 아니라 연구 외적인 요인들 또한 중요하다는 점을 깨닫게 해주셨습니다. 그리고 지난 기간 동안, 짧으나마 연구실을 함께 해왔던 이세린 박사님을 비롯하여 미영 누나, 승은, 상욱에게도 감사의 인사를 전합니다. 특히, 지금까지도 함께 하고 있는 재홍, 세준, 창민, 현구, 은석, 승민, 서현이에게 깊은 감사를 드립니다. 한 치 앞을 내다볼 수 없는 상황에서, 어쩌면 기대할 수 없는 것들까지 대비해야만 하는 악조건 속에서도 이들과 함께 했기에 비록 작지만 절실한 도움과 위로를 받을 수 있었던 것 같습니다. 이 친구들에게도 부디 평안한 미래가 있기를 진심으로 기원합니다.

끝으로 이 기간 동안 힘들게 기다려준 저희 가족들, 여자친구, 그리고 저와 교류했던 많은 선후배 및 동기, 그 외의 여러 친구들에게 감사드리며 이 글을 마칠까 합니다.

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.

(*Youngseo Cha*)

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Many insects rely on their eyes to obtain visual information in an unfamiliar environment. However, unlike mammals, insects have a small brain that consists of few neurons, so they would seem likely to have difficulty processing visual information. Nevertheless, many biological studies have shown that insects can move to specific positions and return to their homes with high probability as they forage for food. This infers that the insect can process visual information efficiently. One of the methods used for efficient processing of visual information is optical flow. Many research studies have demonstrated that some insects control their movement and position with optical flow. Their visual system is attracted by the intensity of optical flow that depends on differences in visual patterns. This chapter deals with vision of insects that use optical flow to navigate their environments.

1.1 A bio-inspired model for robotic navigation with optical flow

Many insect eyes have low resolution, making it difficult to distinguish objects in detail. Nevertheless, insects can maintain their correct positions in unknown landscapes; this infers that insects have biological systems with robust low resolution. Many theories have been proposed to explain the control system of insects. When insects obtain information through visual perception, their locomotion in varying surroundings creates vision patterns that consist of many optical flow vectors (Gibson, 1950; David, 1979; Horn and Schunck, 1981). Many insects, especially those having compound eyes, have many photoreceptors that stimulate the intensity of optical flow (Reinagel, 2001). The vertical system, horizontal system, and tangential neurons of the eye are typical examples. These neurons respond to light stimulation when the optical flow is moved in a specific direction by the observer or by objects in the landscape.

The term of optical flow describes the rate vector of the change of image motion in the retina or a visual sensor that is extracted from the motion of the autonomous agent. This helps insects obtain information from their environment in spite of many different surroundings. The intensity of optical flow is dependent on the distance of the object from the observer and the speed of movement of the object and the observer. Consequently, a viewer using optical flow can obtain much information from these variables. For example, insects control their lateral positions ((Kirchner and Srinivasan, 1989; Srinivasan et al., 1991), speeds (Srinivasan et al., 1996; Baird et al., 2006; Fry et al., 2009), and elevations (Kennedy, 1951; Srinivasan et al., 1996; Baird et al., 2006; Franceschini et al., 2007; Portelli et al., 2010) with optical flow.

Even though insects have little capability to process information in their brains, they generally can navigate with high accuracy across a complex natural environment. The concept of 'optical flow' can be interpreted in a broad sense as involving visual tracking as well as optical flow vectors. Many researchers have suggested several mechanisms to understand insect navigation by optical flow.

Since optical flow was introduced as model of insect vision to use navigation, many approaches have been taken to explain how to use optical flow based on insect vision. The initial idea was developed as guidance methods with local visual homing that refers to the surroundings from an agent. For example, honeybees use landmarks on their navigation routes to calculate optical flow vectors to guide themselves during foraging trips (Collett and Lehrer, 1993). Therefore, if a simple structure of feature detection is given, the difference in landmarks from movement of an agent can be described as a movement of optical flow and can also be used to estimate the movement of the agent. In this paper, we will introduce local visual homing methods to solve a homing navigation task, using a bio-inspired model as a possibility for a low capability memory and sensory system.

Bio-inspired models of robotic navigation based on an optical flow model have advantages of simplicity and robustness when compared with other algorithms. Therefore, proposed models focus on applications of these bio-inspired advantages and on convergence with other fields for robotic navigation. Local visual homing navigation is inspired by the homing navigation observed in insects. Some insects use optical flow systems to find their homing direction within their environments. Even though they do not have complex localization systems, such as place cells and grid cells, in their neurons, they can find their routes with fairly reliable accuracy. They overcome low capability information storage and processing systems by having distinctive mechanisms unlike those found in other systems. For example, instead of using depth information, local visual homing navigation uses intensity information. This navigation falls into several categories depending on the method used for estimation of the homing direction. Moller and Vardy (Möller and Vardy, 2006), classify homing navigation based on intensity information coarsely into correspondence methods and holistic methods. The holistic methods involve image warping, parameter methods, and DID (Descent in Image Distances), while differential flow methods and matching methods are affiliated to correspondence methods. We introduce various homing methods and further investigate bio-inspired homing navigation concepts, such as effective computation, orientation problems, localization, etc.

In the next section, we introduce detailed motivation and objectives for the models proposed in this paper.

1.2 Motivation and objectives

The proposed models of a bio-inspired visual system with optical flow have specific attributes that are useful in the design of robotic navigation. The performance of robotic navigation can be enhanced by adopting bio-inspired models essentially in terms of their robustness and efficiency of computation from the environment. The detailed objectives are as follows:

Accumulated navigation with optical flow Previous research on bio-inspired visual navigation has usually focused on the snapshot model, which is one of the popular approaches used to explain navigation strategies based on honeybee experiments (Cartwright and Collett, 1987). Insects that use a snapshot model take two pictures of their surroundings—one at the starting point and the other at the goal point—through the eyes of the agent and they extract navigation information by matching the two different pictures. Next, they estimate their position and moving distance. However, applications for accumulating the deviation between the two images have not yet been studied. The model proposed here for accumulative flow navigation is based on step-by-step direction recognition of the agent unlike the snapshot model. The direction and state of agent are found using an optical flow algorithm.

Automatic and robust algorithm of the measurement model We proposed average landmark vector (ALV) algorithm (Lambrinos et al., 2000) based on Kanade-Lucas-Tomashi (KLT) tracker (Barron et al., 1994) to design automatic and robust visual system. The ALV has a powerful algorithm for simply and easily estimating the homing direction but this algorithm has two problems that limit the application of the agent in a real environment. First, using automatic object detection based on landmarks to create an optical flow vector is a difficult problem when the surroundings of the agent are too complex to distinguish visual cues with image processing. Second, the ALV algorithm needs external compass information to find an accurate homing direction. The proposed model improves upon previous models that use only visual information.

Probabilistic approaches of local visual navigation A combination of bio-inspired navigation and probabilistic approaches was regarded as an inefficient method for finding a homing direction due to the excessive computational load. The concepts of a bio-inspired model and probabilistic approaches also seem to conflict from the aspect of a trade-off between computational load and robustness of performance. However, optimization of the probabilistic approach with only a one-dimensional input, such as the direction of the homing vector, can give rise to an effective model with this trade-off. Non-parametric filters such as histogram or particle filters, as well as Kalman filters, can help to improve the robustness of performance and to maintain convergence of performance after many steps of the agent. In addition, the localization problem can be also solved with a bio-inspired model based on a low dimensional probabilistic approach. The representative specification of localization inspired biological navigation does not need a pre-searching phase of the agent to recognize a spatial state as prior information. One image taken from the reference position is sufficient for the agent to estimate its own state with high precision.

1.3 Organization of the dissertation

In this chapter, we introduced the motivation, the concept of bio-inspired research, and the objectives of vision-based homing robot navigation methods, which we propose and investigate in this paper. The background underlying the proposed model is shown in Chapter 2, which introduces optical flow, a local visual homing navigation algorithm, and the Bayesian approach.

Chapter 3 presents the accumulative method with optical flow patterns and demonstrates the homing direction of the agent from patterns of optical flow in omni-directional camera, based on the snapshot hypothesis. The accumulative method with optical flow can estimate the movement of the agent with a homing direction. Path integration and homing navigation can be activated to use this method.

An automatic ALV method based on the KLT algorithm and a new visual compass are introduced in Chapter 4. The proposed method overcomes the previous problems of ALV, such as the difficulty in recognizing objects, the speed of image processing, and the visual compass. The ALV method based on KLT is one of the practical approaches based on bio-inspired navigation for improved precision and speed. Flow search as a visual compass is proposed in this chapter and uses only some points on the image, whereas previous methods for a visual compass use all or almost all of the points on the image. Despite the reduction in the amount of information, flow search gives a better performance for finding the alignment of the agent than do previous visual compass methods, such as linear search and phase correlation approaches.

Chapter 5 investigates a combination of probabilistic approaches and the bio-inspired visual navigation model. The computational load of probabilistic approaches is overcome by converting the input data of the probabilistic model to one-dimensional information as the direction of the homing vector. This is similar to the place cells and grid cells in the mammalian brain and creates the desired map made from specifications of homing navigation with optical flow. The vectors are made with centralized directions from the reference position, providing conclusive evidence for application of localization without a pre-searching phase to recognize the spatial state on the map as well as to provide homing navigation. Therefore, of the use of probabilistic models such as parametric and non-parametric filters can help to improve performance of visual navigation.

The final Chapter presents conclusions about the proposed model. The importance of the proposed model and future work are discussed and the model is compared with models from previous research.

Chapter 2

Background

Local visual homing navigation is inspired by the homing navigation used by insects. Some insects use an optical flow system to find their homing direction within their environment. Insects lack complex localization systems such as place cells and grid cells in their neurons, but they are able to find their routes with fairly reliable accuracy. They adapt to their low capability for information storage and processing by the use of distinctive mechanisms that differ from other natural systems. Instead of using depth information, the local visual homing navigation uses intensity information and is based on a bioinspired approach. This navigation can fall into several categories depending on the method used for estimation of the homing direction. Homing navigation based on intensity information can be coarsely classified into correspondence methods and holistic methods (Möller and Vardy, 2006; Möller, 2009; Möller et al., 2010). Holistic methods can use image warping, parameter methods and DID (Descent in Image Distances), whereas differential flow and matching methods are affiliated with correspondence methods.

2.1 Local visual homing methods

2.1.1 Overview

Simultaneous localization and mapping (SLAM) uses complex probabilistic approaches (Milford et al., 2004, 2006; Sunderhauf and Protzel, 2010; Milford and Wyeth, 2008), Moller's group (Möller and Vardy, 2006; Möller, 2009; Möller et al., 2010), proposes

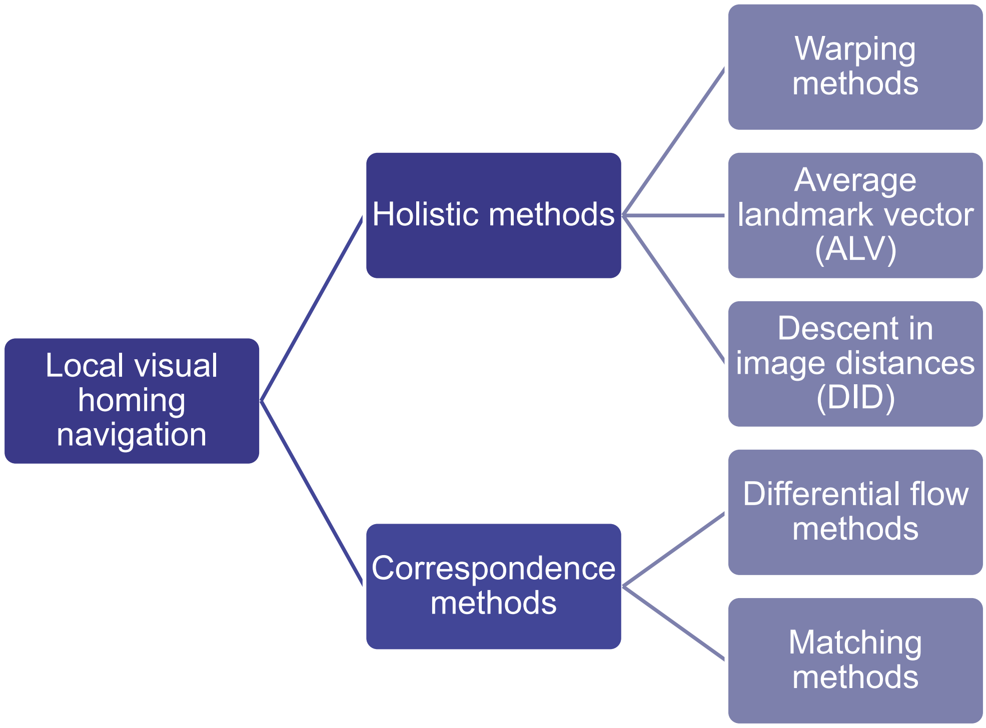


Figure 2.1: Classification of local visual homing methods (Modified from Möller and Vardy (2006))

that a classification of local visual homing methods into two approaches in accordance with the types of information used, such as depth and intensity information. Figure 2.1 shows the broad classification of local visual homing methods (Möller and Vardy, 2006). However, since homing methods using depth information require several sensors and complex strategies and algorithms (Stürzl and Mallot, 2002; Franz et al., 2008), we focus on the intensity information based on the change of image pixels from the camera rather than depth information.

Methods that use intensity information incorporate several ideas for smooth estimation of homing direction. The snapshot model is one of the main ideas inspired by insect navigation. The snapshot model refers to the way that insects store a "snapshot image " containing environmental information about their home position and use this to return to their home by comparing their current image and their home image at an arbitrary position (Cartwright and Collett, 1983; Wehner and Räber, 1979; Wehner et al., 1996; Wehner, 2003). Much evidence supports the use of snapshot matching by some insects, such as ants (Wehner and Räber, 1979; Harris et al., 2007), bees (Cartwright and Collett, 1983, 1987) and wasps (Zeil, 1993). Finding the differences in pictures relies on visual landmarks, which can play an important role as a reference for the

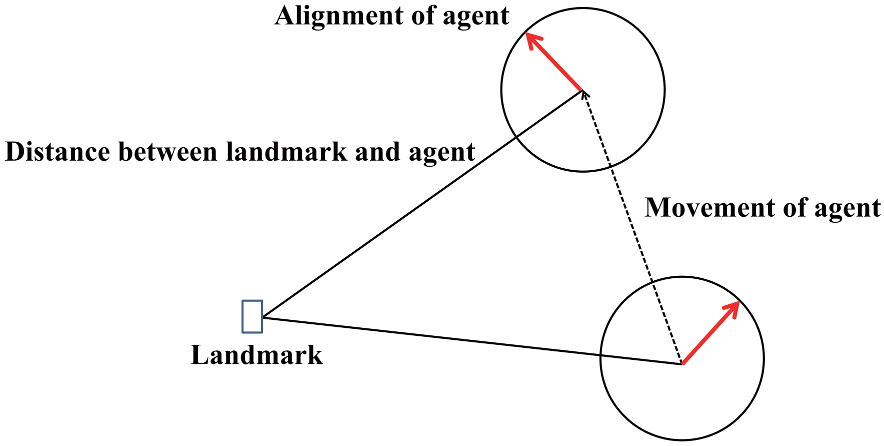


Figure 2.2: Estimation of homing direction of agent when agent moves from home position to current position (Modified from Franz et al. (1998))

optical flow vector on the obtained images. In an actual application, the agent stores only one image from the home location and calculates the difference in optical flow between two images captured at the home and current locations. Agent can be defined as observer to estimate spatial state with measurement updating.

Local visual homing that uses intensity information does not use depth information; therefore, this method relies on the assumption that all features or landmarks on the surroundings are found at roughly the same distance from the agent. This assumption is referred to as the equal distance assumption (Franz et al., 1998).Even though each feature on image has a different respective distance, this assumption does not critically affect the performance of the estimation of the homing direction and can help to simplify the calculation without requiring an additional system to measure the distance to each feature.

Two sub-methods are available for homing navigation using intensity information: the holistic method and correspondence method. In holistic methods, the agent uses the image as a whole to estimate state. On the other hand, the correspondence method focuses on certain points such as corners, which are easy to use for tracking movement between two images.

2.1.2 Holistic methods

2.1.2.1 Warping methods

Warping methods were first introduced by Franz ‘s group (Franz et al., 1998). As shown in Figure 2.2, the agent makes warping sets of the image to correspond to certain movements, which the agent can choose as possibilities.T he warping image refers to an image taken from the home location that is distorted by estimating the image toward the direction of movement of the agent. A matching position is selected using the Euclidean distance from each pixel between the reference and current image. If the state of minimum distance is found, this state can be the real state of the agent after moving from the homing position. The advantages of warping methods are general robustness and the combined processing of the visual compass.

Many different versions have been developed since the first version of the warping method. Stürzl et al. used the calculation distance with warping methods on the Fourier axis (Stürzl and Mallot, 2006; Stürzl and Möller, 2007). A disparity map taken between two images also can be used as a warping approach (Franz et al., 2008). his idea provides an application model warping method using Kinect or a laser sensor that can build a disparity map more quickly than can a model that uses pure visual images. Improvements in performance have been suggested by extending the 2-dimensional version of warping (Möller, 2009; Möller et al., 2010; Labrosse, 2007). Many other application versions have been proposed (Argyros et al., 2001; Gaussier et al., 2000; Argyros et al., 2005; Goedemé et al., 2005; Adorni et al., 2001).

2.1.2.2 Average landmark vector(ALV) methods

The original ALV is one of the parameter methods created by Lambirinos et al. (Lambrinos et al., 2000). The ALV has a powerful performance and is an intuitive algorithm. The ALV consists of two phases to find the homing direction: feature extraction and calculation of difference between two average landmark vectors from each location of the agent. In the feature extraction phase, we can choose several approaches to extract features according to the environment. If the environment consists of simple segmentation, we can choose a binary algorithm or a clustering algorithm such as MEANshift (Comaniciu and Meer, 2002) or CAMshift Bradski (1998). The right hand image in Figure 2.3 shows an example of simple clustering with a binary algorithm. However, if

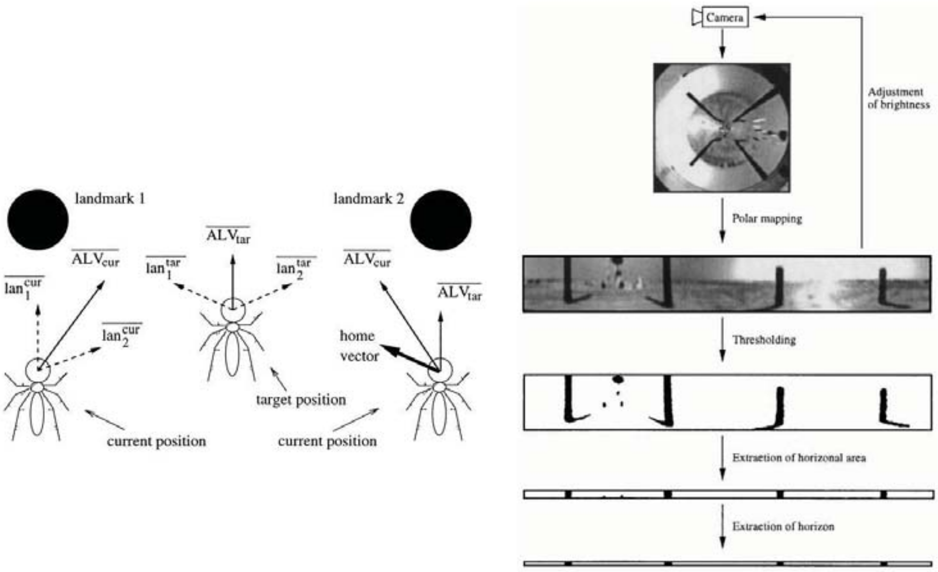


Figure 2.3: Calculation of homing direction with average landmark vectors(left image), example of feature extraction from environment to make landmark(right image) (Reprinted from Lambrinos et al. (2000))

the environment is very complex so that finding marked segmentations is not possible, the clustering algorithm is a useless approach because this algorithm cannot distinguish the main factors on the image in a complex environment. This task can be resolved by choosing corner detection as an alternative method essentially as described by Harris and Stephens (Harris and Stephens, 1988). These authors propose that the corner can be defined as a point containing a large differential value to the perpendicular direction of another. To describe this condition in mathematical terms, the eigenvalue of the autocorrelation matrix from a certain position helps to distinguish whether this position is a corner or not. A higher eigenvalue gives a sharper difference between the original pixel and neighboring pixels.

An autocorrelation matrix can be derived as the following:

$$A = \begin{pmatrix} \sum w_{i,j} I_x^2(x+i,y+j) & \sum w_{x,y} I_x(x+i,y+j) I_y(x+i,y+j) \\ \sum w_{i,j} I_x(x+i,y+j) I_y(x+i,y+j) & \sum w_{i,j} I_y^2(x+i,y+j) \end{pmatrix} \tag{2.1}$$

To pick out better corners with this concept, Shi and Tomashi (Shi and Tomasi, 1994) suggest a threshold of eigenvalues on an autocorrelation matrix. Between two eigenvalues, if the smaller eigenvalue is larger than a certain threshold, this point is a good feature to track. This idea also is applied to the Kanade-Lucas-Tomashi tracker (KLT

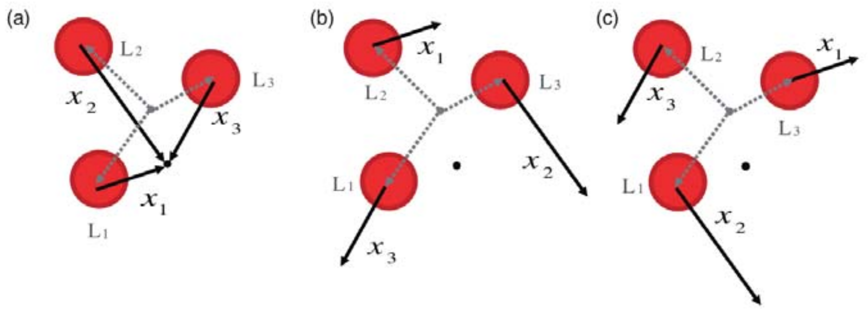


Figure 2.4: DELV mechanism using several remarkable landmarks (Reprinted from Yu and Kim (2011b))

tracker) (Barron et al., 1994) to measure differential flow between two images. This part will be introduced in a subsection of differential flow methods. For preferential calculation of the homing direction phase, the average landmark vector at a certain position is calculated. The left hand image in Figure 2.3 indicates that desert ants calculate a vector from two landmarks at their current position. The simple equation of calculation for an average landmark vector can be described as the following:

$$\begin{aligned} \overline{ALV_{tar}} &= \sum_{i=1}^{N} \overline{lan_i^{tar}} \\ \overline{ALV_{cur}} &= \sum_{i=1}^{N} \overline{lan_i^{cur}} \end{aligned} \tag{2.2}$$

where N is the number of measured landmarks at a certain position. From the equal-distance assumption, the size of each vector can be regarded as a unit, so the calculation of the homing vector is

$$\overline{h} = \overline{ALV_{tar}} - \overline{ALV_{cur}} \tag{2.3}$$

where $\overline{h}$ is the homing vector.

The advantages of ALV are simple and intuitive to apply to other navigation tasks, but this algorithm requires an external compass and is closely dependent on the landmark state (Möller, 2000; Möller et al., 2001). For instance, if part of the landmarks are occluded, this can drastically degrade the performance for finding the homing direction (Angulo and Godo, 2007; Smith et al., 2007; Moller, 1999).

The representative variations of ALV are cases used on a Fourier axis (Menegatti et al., 2004) and a distance-estimated landmark vector(DELV) method that can find the homing direction without an external compass to guess distance from landmarks in the environment (Yu and Kim, 2011a,b). Figure 2.4 shows that DELV can automatically find

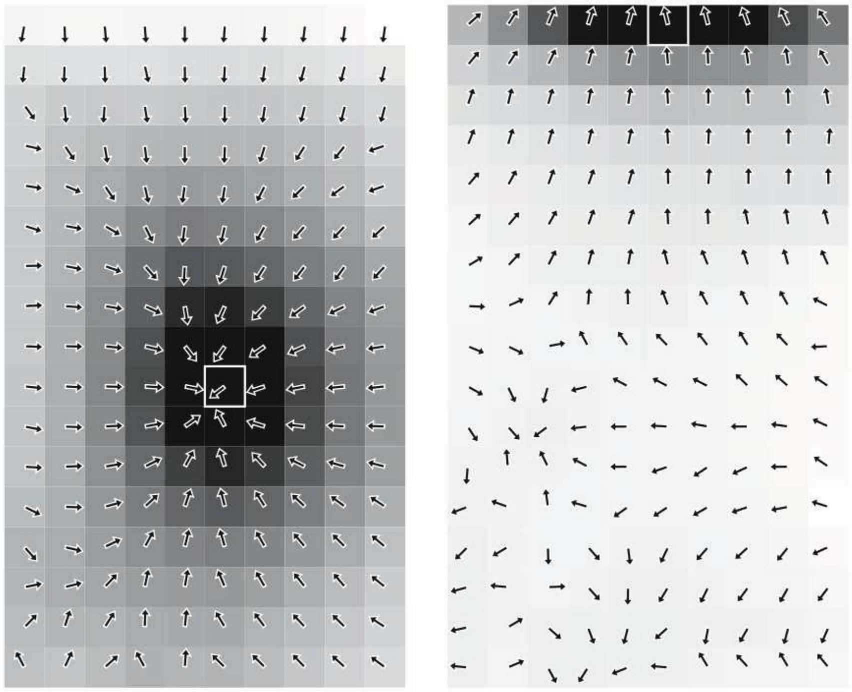


Figure 2.5: Examples of DID gradient distribution on the Original data set from *www.ti.uni-bielefeld.de/html/research/avardy* (Reprinted from Möller and Vardy (2006)): image on the left side has its homing position at (5, 7) and the image on right side has its homing position (5, 16)

the optimal arrangement when several landmarks are given.

2.1.2.3 Decent in image distances(DID) methods

The DID methods are presented by Zeil et al. (Zeil et al., 2003, 2009). These methods calculate the distance between two original images and obtain a gradient of movement for the current position to the home position (Stürzl and Zeil, 2007). This idea is based on the assumption that an image taken from a location near the home position varies smoothly and monotonically with spatial distance when compared with an image taken from the home position (Zeil et al., 2003; Möller and Vardy, 2006). These methods can therefore estimate the homing direction without an external compass.

Figure 2.5 shows examples of the DID gradient distribution for drawing vectors toward the home location. In general, the performance is better when the home position is located at center than at the edge of the map due to conservation of visual information. When the home position is at the center, this gives a high possibility that it

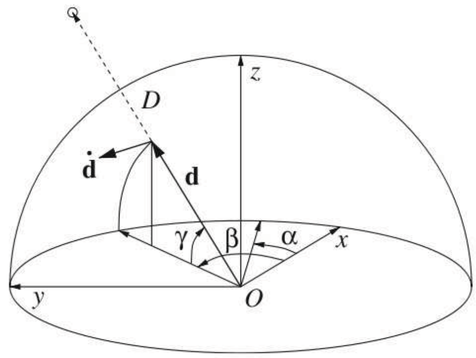


Figure 2.6: Omni-directional camera lens described in spherical coordinates (Reprinted from Möller and Vardy (2006)).

contains omnipresent visual information. Therefore, the gradient of the image to the home location can be estimated precisely in the center case. This implies that the DID assumption can be obeyed when visual information is abundant and that DID imposes a limitation on the range distance from the home position. To resolve this problem, the way-point method was introduced and is described in detail in Section 3.

To describe this method mathematically, we use the sum squared error (SSE) used by Möller and Vardy (Möller and Vardy, 2006) instead of the RMS method used Zeil et al. (Zeil et al., 2003, 2009) because even if these operators are not completely the same, they have similar meanings to evaluate distances and the SSE operator can be easier to describe than the RMS operator is. When C is current image and S is the snapshot image taken from a reference position, the SSE can be defined by

$$SSE(\mathbf{x}) = \frac{1}{2}\sum_{i,j}[C(\varphi_{ij},\mathbf{x}) - S(\varphi_{ij})]^2 \tag{2.4}$$

where φ means a variable on the spherical coordinates and $\mathbf{x}$ is a position vector of the agent on the Cartesian coordinates. Figure 2.6 shows some components of the spherical coordinates.

Estimation of the homing direction requires a negative gradient operator. When $h(\mathbf{x})$ is the homing vector, the equation is given as the following:

$$h(\mathbf{x}) = -\nabla_{\mathbf{x}} C(\varphi_{ij},\mathbf{x}) \sum_{i,j}[C(\varphi_{ij},\mathbf{x}) - S(\varphi_{ij})]^2 \tag{2.5}$$

However, the $\mathbf{x}$ term do not help to calculate the homing vector because the camera image consists of φ a spherical coordinate by an omni-directional camera. To resolve this problem, we can change $\mathbf{x}$ to φ by using an additional assumption that the value of a pixel does not change if the movement of the camera image and the agent is very small. The mathematical version of this assumption is as follows:

$$C(\varphi+\Delta\varphi, \mathbf{x}+\Delta\mathbf{x}) = C(\varphi, \mathbf{x}) \tag{2.6}$$

In addition, $C(\varphi+\Delta\varphi, \mathbf{x}+\Delta\mathbf{x})$ can be approximated to terms $C(\varphi, \mathbf{x})$ by using Taylor expansion as follows:

$$C(\varphi+\Delta\varphi, \mathbf{x}+\Delta\mathbf{x}) \approx C(\varphi, \mathbf{x}) + \nabla_\varphi C(\varphi, \mathbf{x})\Delta\varphi + \nabla_\mathbf{x} C(\varphi, \mathbf{x})\Delta\mathbf{x} \tag{2.7}$$

We can derive an equation to express the spatial relation and intensity gradient as the following:

$$-\nabla_\varphi C(\varphi, \mathbf{x}) \approx \nabla_T C(\varphi, \mathbf{x})\Delta\mathbf{x} \tag{2.8}$$

We find the relationship of the optical flow model using this modeling (Möller and Vardy, 2006; Möller et al., 2007; Möller, 2012) derived from the Koenderink-van Doorn flow equation (Koenderink and Doorn, 1987) to compose the equation of the proposed algorithm. These authors describe an optical flow vector with two parts, which are translational and rotational components, as the following:

$$\mathbf{v} = \frac{\mathbf{x} - (\mathbf{x}^t v)v}{D} - R \times v \tag{2.9}$$

where $\mathbf{v}$ is optical flow vector. $\mathbf{x}$ and R are the translational and rotational components, respectively. D is the distance between the camera and a certain point of the scene. v is the movement of a feature for the movement of the camera.

If the agent with the camera has a compass to align the angle, because the movement of the agent is affected by noise from external factors without a reference direction, the rotational component R can be neglected with zero. can be neglected as zero. However, if the agent does not have a compass or if it has rotational movement as well as pure translational movement, in that case, R is not zero so the optical flow vector can be distorted by complex factors. The issue of minimizing the rotational component for accurate estimation is discussed in section 3 with an additional algorithm. Therefore, in this model, we consider the rotational component R to be zero.

According to Möller et al. (Möller and Vardy, 2006; Möller et al., 2007; Möller, 2012), Translational component $\mathbf{x}$ can be described with x, y for camera movement using a component of Cartesian coordinates, with direction angle α and speed ν as the following:

$$\mathbf{x} = (x, y, 0)^t = \nu(\cos\alpha, \sin\alpha, 0)^t \tag{2.10}$$

and we define φ as a 2D vector of angles in spherical coordinates and we also rewrite $\mathbf{v}$ and v as aspect of the spherical coordinates, $\varphi = (\beta, \gamma)$ and $\dot{\varphi} = (\dot{\beta}, \dot{\gamma})$, respectively. β is the horizental angle and γ is the vertical angle for angular speed vector. From this concept, we obtain the equation that transforms Cartesian coordinates to spherical coordinates as the following:

$$\begin{pmatrix} \dot{\beta} \\ \dot{\gamma} \end{pmatrix} = \frac{\nu}{D(\varphi)} \begin{pmatrix} \sec\gamma & 0 \\ 0 & \sin\gamma \end{pmatrix} \begin{pmatrix} \sin\beta & -\cos\beta \\ \cos\beta & \sin\beta \end{pmatrix} \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} \tag{2.11}$$

In addition, if $(\cos\alpha, \sin\alpha) = \nu(\dot{x}, \dot{y})$ is ture from equation 2.10, we set the equation that expresses the changes in the movement of features of the environment to flow on the spherical camera as the following:

$$\dot{\varphi} = \frac{1}{D(\varphi)} \begin{pmatrix} \sec\gamma & 0 \\ 0 & \sin\gamma \end{pmatrix} \begin{pmatrix} \sin\beta & -\cos\beta \\ \cos\beta & \sin\beta \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \tag{2.12}$$

If the camera has a small translational movement of $\Delta\mathbf{x} = (\Delta x, \Delta y)^t$, equation 2.12 can be approxmated as the following:

$$\Delta\varphi = \frac{1}{D(\varphi)} \begin{pmatrix} \sec\gamma & 0 \\ 0 & \sin\gamma \end{pmatrix} \begin{pmatrix} \sin\beta & -\cos\beta \\ \cos\beta & \sin\beta \end{pmatrix} \Delta\mathbf{x} \tag{2.13}$$

From this equation 2.8, we substitute $\Delta\varphi$ to terms from equation 2.13 as the following:

$$\nabla_{\mathbf{x}} C(\varphi, \mathbf{x}) \approx -\frac{1}{D(\varphi)} \begin{pmatrix} \sec\gamma & 0 \\ 0 & \sin\gamma \end{pmatrix} \begin{pmatrix} \sin\beta & -\cos\beta \\ \cos\beta & \sin\beta \end{pmatrix} \nabla_{\varphi} C(\varphi, \mathbf{x}) \tag{2.14}$$

This gives us the home vector function $\nabla_{\varphi} C(\varphi, \mathbf{x})$ instead of $\nabla_{\mathbf{x}} C(\varphi, \mathbf{x})$ from equation 2.5 as the followings:

$$\begin{aligned} h(\mathbf{x}) = -\sum_{i,j} \frac{1}{D(\varphi_{ij})} \begin{pmatrix} \sec\gamma_j & 0 \\ 0 & \sin\gamma_j \end{pmatrix} \begin{pmatrix} \sin\beta_i & -\cos\beta_i \\ \cos\beta_i & \sin\beta_i \end{pmatrix} \nabla_{\varphi_{ij}} C(\varphi_{ij}, \mathbf{x}) \\ \times [C(\varphi_{ij}, \mathbf{x}) - S(\varphi_{ij})] \end{aligned} \tag{2.15}$$

However, based on the equal-distance assumption, we can rewrite $D = D(\varphi)$ by accepting an additional assumption that all distances are the same in the image.

The variations of the DID method use Taylor approximation cases such as the Matched-filter DID method(Möller and Vardy, 2006), Newton-based Matched-filter DID method (Möller et al., 2007) and are based on image distortion methods (Binding and Labrosse, 2006; Labrosse, 2007).

2.1.3 Corresponding methods

2.1.3.1 Differential flow method

Basically, differential flow methods are part of the methods for extraction of features and matching correspondence methods. Differential flow is calculated using optical flow algorithms in image processing. These algorithm are based on time-varying images and have a brightness constraint assumption (Fennema and Thompson, 1979; Horn and Schunck, 1981; Nagel, 1982) or a gradient of constraint assumption (Brox et al., 2004, 2009; Papenberg et al., 2006) which means that the brightness of a patch on the image surface does not change when an observer moves to the different position (Hatzitheodorou et al., 2000). However, the optical flow equation has a critical problem; namely, that it has two unknowns but only one equation. This causes infinite solutions for a specific phenomenon; that is, it is an aperture problem. This problem is resolved by confining the number of solutions, so that many optical flow algorithms add an assumption to make a solvable model from the basic equation. Among these algorithms, we decided to use the Kanade-Lucas-Tomasi (KLT) tracker based on Lucas-Kanade algorithm. Details of this optical flow algorithm are provided in next section.

The KLT consists of two parts that extract corners and performs corner matching to estimate position with the optical flow algorithm. The problem whereby a singular matrix cannot become an inverse matrix or differentiability problem is overcome by using a Gaussian filter in preprocessing. Formation of a relationship between the results of KLT tracker and the homing direction function $h(\mathbf{x})$,requires a combination of equations. The aperture problem aggravates to one solution from the LK algorithm, which means that we can first select one solution by choosing the flow vector $\Delta\varphi$ parallel to the intensity gradient as follows (Beauchemin and Barron, 1995):

$$\Delta\varphi = \frac{\nabla_\varphi C(\varphi, \mathbf{x})}{\| \nabla_\varphi C(\varphi, \mathbf{x}) \|^2}[S(\varphi) - C(\varphi, \mathbf{x})] \tag{2.16}$$

The gradient parts of the home vector equation can be changed to terms of the Intensity variable I in KLT. $\| \nabla_\varphi C(\varphi, \mathbf{x}) \|^2$ s a constant value so we can neglect this term due to D which we can set freely from the equal-distance assumption. This modification of the equation allows us to make a simple change to the home vector function to remove $\Delta\varphi$. The result is

$$h(\Delta\varphi_{ij}) \approx -\sum_{i,j} \begin{pmatrix} \sec\gamma_j & 0 \\ 0 & \sin\gamma_j \end{pmatrix} \begin{pmatrix} \sin\beta_i & -\cos\beta_i \\ \cos\beta_i & \sin\beta_i \end{pmatrix} \Delta\varphi_{ij} \tag{2.17}$$

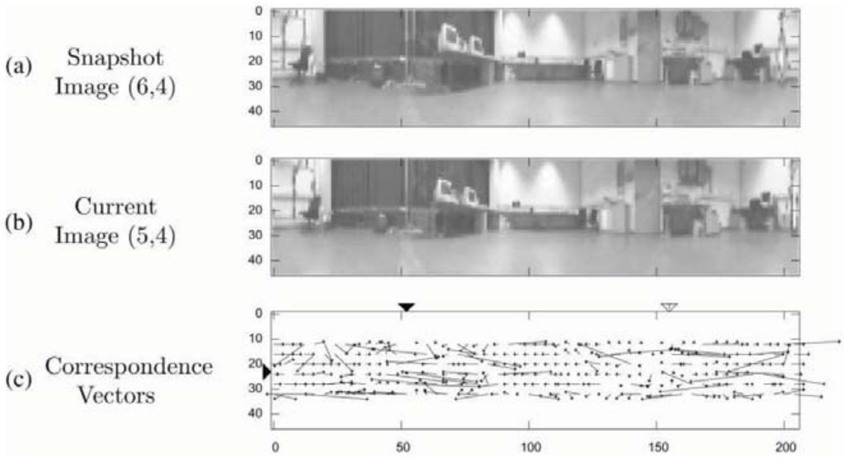


Figure 2.7: Example of block matching method based on snapshot model (Reprinted from Vardy and Möller (2005); Möller et al. (2008)): (a) is snapshot image taken from home location. (b) is current image. (c) is result image of correspondence vectors from (a) to (b).

Equation 2.17 indicates that the home vector can be obtained by differential flow methods, so v calculated from the KLT tracker is valid, as is $\Delta\varphi$, when the home vector is needed. We can organize equation 2.17 as the following:

$$h(\Delta\beta,\Delta\gamma)_x \approx -\sum[\frac{\sin\beta}{\cos\gamma}\Delta\beta - \frac{\cos\beta}{\cos\gamma}\Delta\gamma] \tag{2.18}$$

$$h(\Delta\beta,\Delta\gamma)_y \approx -\sum[\sin\gamma\cos\beta\Delta\beta + \sin\gamma\sin\beta\Delta\gamma] \tag{2.19}$$

From equation 2.18, the homing direction can be calculated by x and y from the images. The advantages of differential flow methods are their moderate performance and fast processing speed (Vardy and Möller, 2005). Controlling the number of corners with the Shi and Tomasi criteria can reduce processing time by removing trivial points on the image (Shi and Tomasi, 1994). On the other hand, a drawback of differential flow methods is that performance in finding the homing direction is inferior to that of the block matching or warping methods.

2.1.3.2 Block matching method

Block matching is one of the optical flow algorithms used to estimate the movement of features on an image (Jain and Jain, 1981). Unlike other algorithms, block matching is based on the matching of the same pixels. By finding the maximum similarity of a

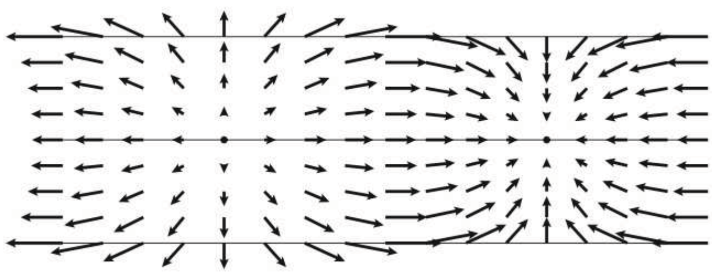


Figure 2.8: Ideal case of corresponding vectors made from snapshot model with optical flow algorithm (Reprinted from Vardy and Möller (2005)): FOC and FOE are located on the converge and diverge point on image, respectively.

group of pixels in a window, the features for tracking can be corresponded to other pixels. The representative measurement models of similarity are RMS, SSE, SSD, SAD, etc.

Figure 2.7 demonstrates an example of block matching to find the homing direction. Many correspondence vectors can be drawn on the image in Figure 2.7. The directional tendency of corresponding vectors is identified using the concepts of *focus of contraction* (FOC) and *focus of expansion* (FOE). The effectiveness of the distortion of the omni-directional camera means that the optical flow vectors on the lens are not uniformly distributed and are distorted by the curved space. This specification causes the position of the correspondence vector distribution to diverge and converge.

The FOE and FOC indicate the relative direction of the agent in the environment. Since a reflection of optical flow on the camera is opposite to the movement of the agent, the movement vector can be regarded as being in the direction from FOC to FOE. Even though a difference in the distance from all the features from the camera causes distortion in the length of movement, the direction can be precisely surmised because independence of direction can be maintained with a different distance from the markers on the image. Figure 2.8 shows an ideal case of corresponding vectors made from a snapshot model with an optical flow algorithm.

Assume that $\mathbf{p}$ and $\mathbf{p}'$ are set of points on first image and that candidate points on the second image match with the local optimization appearing in local regions. We can set the SSD algorithms as the following:

$$SSD(\mathbf{p},\mathbf{p}') = \sum_{i=-r}^{r}\sum_{j=-r}^{r}[S(p_x+i,p_y+j)-C(p'_x+i,p'_y+j)]^2 \tag{2.20}$$

where r is the size of window. The speed of processing can be accelerated by choosing

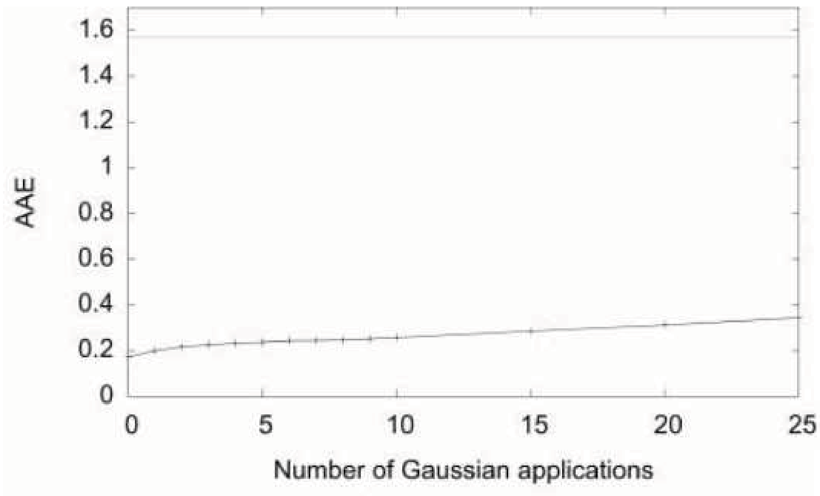


Figure 2.9: Relationship between AAE and the number of gaussian applications (Reprinted from Vardy and Möller (2005))

a small value for r. Vardy and Möller (Vardy and Möller, 2005), choose one as the value of r . From this result, $\check{\mathbf{p}}$ which is the optimal position, can be selected by minimizing SSD, as follows:

$$\check{\mathbf{p}} = \arg \min_{\mathbf{p}' \in E_q(\mathbf{p})} SSD(\mathbf{p}, \mathbf{p}') \tag{2.21}$$

and E_q is as follows:

$$E_q(p_x, p_y) = [(p_x + i, p_y + j) \mid i, j \in \mathbb{R}, \mid i \mid \leq q \wedge \mid j \mid \leq q] \tag{2.22}$$

According to Vardy and Möller (Vardy and Möller, 2005), the block matching algorithm has the best performance compared to other methods. *IntMatch* and *GradMatch* were also introduced to improve processing speed but they are not superior to the block matching algorithm in terms of performance (Vardy and Möller, 2005). In addition, the block matching algorithm is robust for complex scenes where finding landmark features is difficult with differential flow methods.

Figure 2.9 shows the robustness for Gaussian noise. The average angular error (AAE) is one of the representative parameters used to evaluate the performance of the homing vector. The AAE focuses on the difference between the ideal and measurement angle of the homing vectors. If the ideal homing vector is $\mathbf{v_{ideal}}$ and the measurement angle from certain method is $\mathbf{v_{method}}$, AAE can be described as the following:

$$AAE = \frac{1}{mn-1} \sum_{c_x=0}^{m-1} \sum_{c_y=0}^{n-1} \arccos(\mathbf{v_{ideal}}, \mathbf{v_{method}}) \tag{2.23}$$

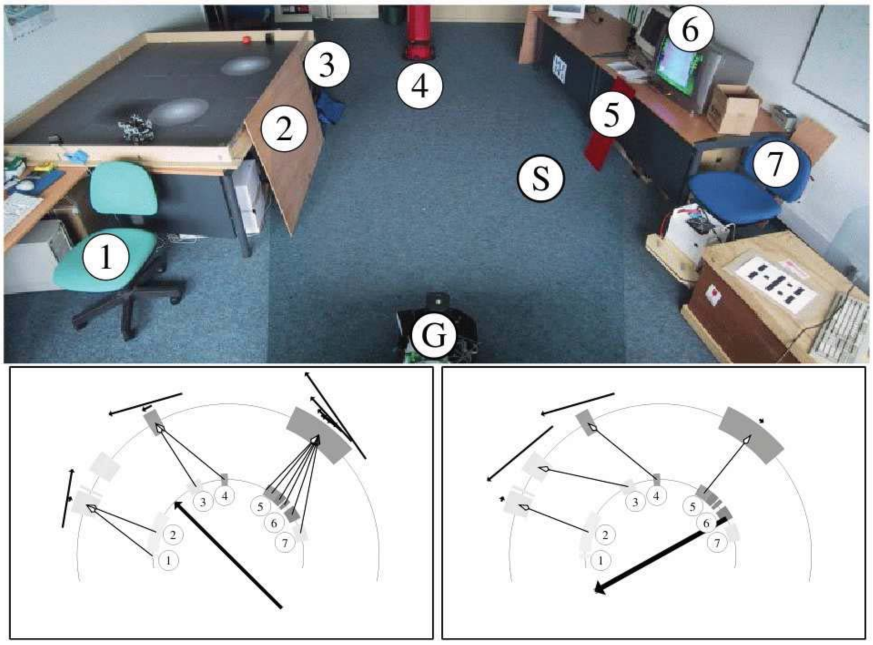


Figure 2.10: Example of color segmentation approach (Reprinted from Gourichon et al. (2002))

where *m* and *n* are the width and height for the number of positions, respectively. Therefore, decreasing the AAE implies a good performance for homing for the agent.

A greater number of Gaussian applications increases the changes that can be made in a dim image. However, AAE is increased gradually for Gaussian noise, as shown in Figure 2.9. This means that the block matching method not only focuses on some matching features but also on the inclination of the whole background.

2.1.4 Other methods

Vardy and Oppacher suggest methods that match the algorithm based on corner detection (Vardy and Oppacher, 2003, 2004, 2005). This case uses a ring operator to match two sets of features from two images. Color segemtation (Gourichon et al., 2002) or color momentum information (Goedemé et al., 2004) can also help to find the homing direction. Figure 2.10 demonstrates processing of color segmentation from the general environment. Color pixels give more information about finding the homing direction than does grayscale information, but the variant features of color still create problems with mismatches between the two images.

The quality of feature matching can be increased by using a scale invariant feature(SIFT) algorithm, which allows the agent to set good landmarks to track (Pons et al., 2007; Ramisa et al., 2011; Luke et al., 2005). The SIFT algorithm can choose features on the image through dynamic distortion of the image to find robust corners that do not change under various conditions. However, due to variations in the conservation of features to track, these algorithms cannot guarantee good performance when compared with other methods.

Rather than engineering performance, the bio-inspired type has recently become a focus through mimicking the behavior of an ant as the predicted model (Möller, 2012). Before it starts moving, the agent observes some of the possibilities for movement of a picture according to the direction of navigation (Baddeley et al., 2012, 2011). From the FOE & FOC model, the tendency of the movement direction can be determined with specific regulations for each direction case. This method can increase the processing speed to save time when using a DID algorithm such as RMS or SSD.

2.2 Applicational issues

2.2.1 Overview

As part of the additional issues, we cover subsidiary issues for the main methods used to find a homing direction, such as orientation, removing noise, and waypoint problems. These issues are also very important because the additional condition of local visual homing navigation directly affects performance.

2.2.2 Orientation methods

Insects use compass information derived from external or internal cues (Wehner, 1989, 1997). Needless to say, an orientation task is a very crucial problem, and some approaches require an external compass, such as ALV, differential flow, and block matching methods. Even if an external compass is provided, an indoor case that cannot use GPS sensor will still need an internal compass. In this section, we cover the visual compass problem where no angle information is available. This problem has been resolved with the development of several solutions.

Labrosse suggests a linear search method to align the angle of the agent (Labrosse,

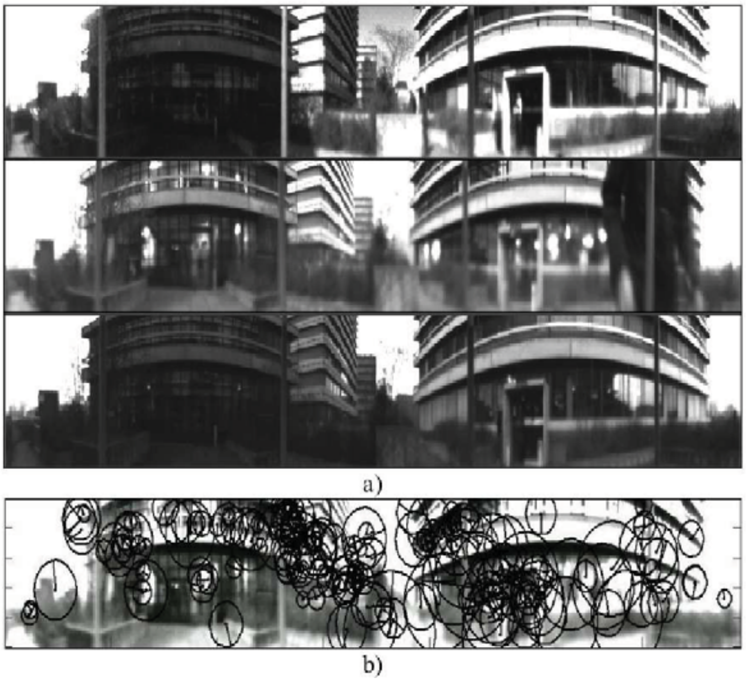


Figure 2.11: Using SIFT algorithm and finding features on image (Reprinted from Pons et al. (2007))

2004, 2006). The linear search method is based on the assumption that the pixels of an image vary smoothly and monotonically with spatial distance (Zeil et al., 2003). This method is simple: Pixels of two images at the same position are subtracted and the difference is accumulated until all of the pixels are subtracted to get a total Euclidian distance for all angles. At a certain angle position with a minimum distance error, we can find the distorted angle from the reference angle.

Burke and Vardy improved the processing speed by proposing two methods – phase correlation and the sample search approach (Burke and Vardy, 2006). Phase correlation, which was created by Kuglin and Hines. uses the axis of a Fourier transform and compares the similarity of patterns between two images. Sample search, in contrast, is a variation of linear search where processing is improved by choosing a minimum position that is faster than the original version. Even though the Fourier transform is a complex computation, the DSP processor helps to accelerate the processing speed. The phase correlation and sample search methods cannot outperform linear search in several tests but the processing time can be rapidly reduced.

Saez-Pones et al. built an orientation model with the SIFT algorithm as well as a local visual navigation model (Pons et al., 2007). Since the corners from the SIFT algorithm have robustness from assorted noise and distortions of the image, the features can be regarded as landmarks just like celestial markers such as the sun or moon. Figure 2.11 shows an example of finding features after image processing. The SIFT algorithm also provides the expectation of matching to another image as well as finding good features to track.

Choosing an optical flow algorithm from differential flow methods is a good choice for performing an orientation, even though flow search requires more processing time than does a general linear search. Basically, the angle distribution is circular and periodic, so bias of color to black or white can mean good performance.

However, although this method has a broad application field (Vachhani and Sabnis, 2011; Montiel and Davison, 2006; Sturm and Visser, 2009; Bellotto et al., 2008; Labhart and Meyer, 2002; Frier et al., 1996), all of the methods of orientation have limitations for range size. If the agent goes out beyond the range, this performance is rapidly degraded. Therefore, a waypoint method is required that can extend range of navigation to avoid this problem.

2.2.3 Removing noise methods

The random sample consensus (RANSAC) approach can help to find the main tendency of the vector direction (Fischler and Bolles, 1981; Meer et al., 1991; Bolles and Fischler, 1981). If landmarks are not given or are difficult to find initially in the complex environment, the use of a feature detection algorithm is inevitable for processing. Unfortunately, all of the features do not have the same relative movement when the agent starts to move to a certain space. However, we can agree that the relative movement shown from most of the features has a high probability of real relative movement for agent movement. Therefore, if a certain algorithm can find a tendency for most of elements, this problem can be solved.

Basically, RANSAC is one of the sampling methods. It means that RANSAC is not always guaranteed to find a global solution. However, if the number of samplings is sufficient, the results of RANSAC can be converged to a global solution. Methods using many features for tracking can be applied by this method to improve the performance of estimating the homing direction.

The LK algorithm has a noise removing filter that is initially in two parts. The first

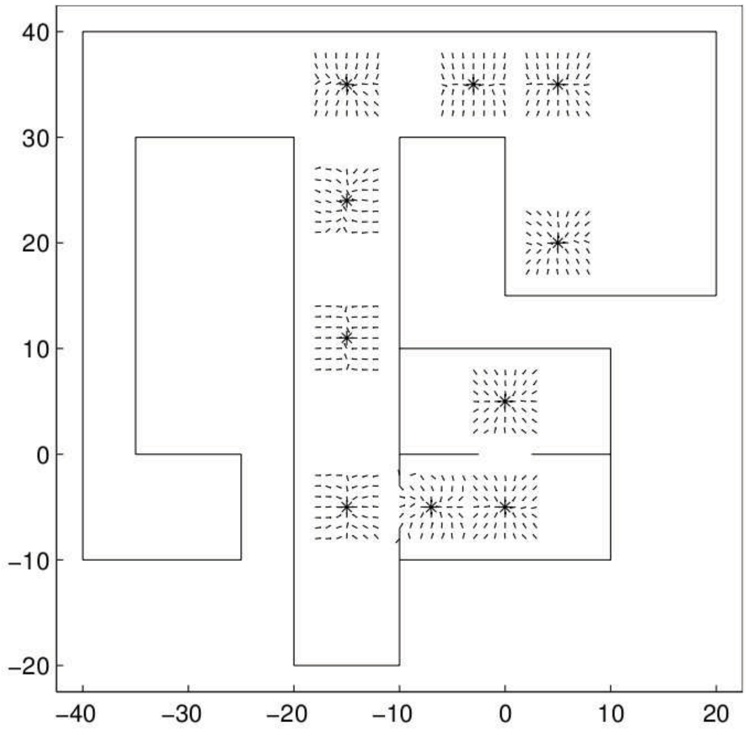


Figure 2.12: Example of waypoint method (Reprinted from Vardy (2006)): Each waypoint has a set of vectors for homing direction. Agent can choose a suitable waypoint for itself by using a matching algorithm like SSD.

part avoids making a singular matrix as $\nabla I^t \nabla I$ from eqaution 2.30 when extracting features. The second part is a pre-calculation of the estimated distance of the vector from each feature. If the distance is over the threshold value, this optical flow vector can be regarded as a noise vector. This idea shows that image matching and orientation based on the LK algorithm can be applied to the general case. This algorithm has a higher performance when compared to the other methods in the moderate range from the home position.

2.2.4 Waypoint method

Local visual homing methods based on vision sensory systems have the limitation that if the agent goes out a certain range from the home position, performance degrades rapidly. Performance of homing requires a waypoint method. Vardy suggested a waypoint method based on a vision system and simulated this method in a specific imaginary space (Vardy, 2006).

Vardy determined the positions of the by setting two parameters; namely, the angle and the distance threshold. The threshold value allows the agent to calculate the moments when a picture can be taken using an omni-directional camera and to estimate whether the current position can be regarded as waypoint. Figure 2.12 shows the process for setting several waypoints on an imaginary map with a complex background. However, each waypoint has some error and can be accumulated by movement that is dependent only on the homing direction. Therefore, if an agent is located near a certain waypoint with the local visual navigation algorithm, the agent can calculate with SSE instead of the general homing navigation algorithm until arriving at the waypoint position. This method reveals the important feature that an agent with local a visual homing navigation algorithm can navigate across a long range area.

2.2.5 RatSLAM

Milford et al. (Milford et al., 2004, 2006; Milford and Wyeth, 2008) suggested a RatSLAM that was inspired by a mammalian place cells. The RatSLAM has two parts: localization and mapping. When the agent encounters a phase of localization, it uses an encoder and image matching in a database to estimate the spatial state. In the RatSLAM, pose cells act as a combination of place cells and grid cells in biology. Pose cells have several spatial state contents, such as position, orientation etc. The agent can compare information from the pose cells with local view cells stored as visual data that the agent obtains by taking a picture at each position. Hebbian learning then shows that these contents have a relationship for estimating the agent's position, just like SLAM.

The specification of RatSLAM uses a particular probabilistic update rather than a general Bayesian update (Sunderhauf and Protzel, 2010). The additive update model of the RatSLAM can help to upgrade the performance of the RatSLAM, but this model incurs a high computational load compared with the general Bayesian update. Computational load can be reduced by using a fusion model for multiplicative and additive updates.

2.3 Optical flow algorithm

2.3.1 Overview

To understand the proposed model, the concepts of the optical flow algorithm must first be introduced. The optical flow algorithm is essentially based on time-varying images and has a brightness constraint assumption (Fennema and Thompson, 1979; Horn and Schunck, 1981; Nagel, 1982) which is that the brightness of the image of a patch on the surface does not change as the observer moves relative to the surface. The approximated Taylor expanding version of the assumption is the equation, as follows:

$$\frac{\partial I}{\partial x}\frac{dx}{dt}+\frac{\partial I}{\partial y}\frac{dy}{dt}+\frac{dI}{dt}=I_x u+I_y v+I_t=0 \tag{2.24}$$

Where I_x, I_y and I_t are each differential variables of intensity and u and v are optical flow vectors on the x and y axis, respectively.

However this equation 2.24 has critical problem which is that there are two unknowns but one equation. Consequently, this case causes infinite solutions for a specific phenomenon; that is, it causes an aperture problem. This problem can be resolved by confining the number of solutions, and many optical flow algorithms add an assumption to make a solvable model from the basic equation.

We chose the LK algorithm to describe the optical flow vectors. The LK algorithm has several well-known strong points: it is intuitive, fast, and easy to use. Another popular optical flow algorithm is the HS algorithm. In spite of the merits of the HS algorithm for global matching, when used with the LK algorithm for local matching, this algorithm has a drawback in that it is too sensitive compared with the LK algorithm to process an erratic brightness environment because the HS algorithm considers brightness by setting theλ factor on this equation as follows:

$$E(u,v)=\iint (I_x u+I_y v+I_t)^2+\lambda(u_x^2+u_y^2+v_x^2+v_y^2)dxdy \tag{2.25}$$

Where u_x and u_y are each differential variables of optical flow x and y vectors, respectively. For equation 2.25, the aim is to minimize the state of energy E by adding λ which the user can set freely. If the λ value is high on equation 2.25, the marginal value of the smooth constraint also has a high threshold for brightness so that the optical flow vectors made from the HS algorithm usually have small sizes. However, the λ factor can be only determined by the global state, so that the image has to undergo

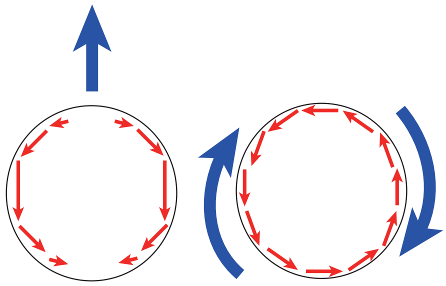


Figure 2.13: Patterns of optical flow on omni-directional camera for translational movement (left) and rotational movement (right)

a pre-processing phase that contains edge detection independent of brightness, such as the Sobel algorithm or the Canny edge algorithm, before applying the HS algorithm(Hatzitheodorou et al., 2000). This process expends a great deal of time and an additional image processing phase could be needed to fix the exact computation of the optical flow. In contrast, the LK algorithm is the basis of local color matching, so brightness is not a main factor when finding optical flow vectors.

An omni-directional camera gets more information about the environment because the angle of visibility for an omni-directional camera is 360 degrees. With the optical flow algorithm, another merit of an omni-directional camera that is perpendicular to all directions is that it avoids visual side effects. For instance, an agent using a general camera cannot observe the optical flow vector for the specific case where the movement direction is parallel to the camera direction. Consequently, an omni-directional camera is a good choice for the optical flow algorithm.

Optical flow consists of two factors: translational and rotational components (Koenderink and Doorn, 1987). Depending on the locomotion types of agents, the pattern of optical flow in a camera will vary. Figure 2.13 shows the main patterns of optical flow for an omni-directional camera for agent locomotion. Translational movement has vector patterns that depend on the movement direction. On the other hand, rotational movement produces vectors with different directions but the same lengths. Rotational vectors are perpendicular to radial directions in the omni-directional camera. By observing the optical flow patterns, we can estimate the spatial information of agent movement, such as distance and direction.

Some limitations are evident for the optical flow system in HS. The high sensitivity of the optical flow system to light causes problems when using a specific indicator that recognizes the surroundings of the observer. The optical flow also has many dependent variables which creates trouble when obtaining information from the surroundings (Koenderink and Doorn, 1987). Sometimes, insects that use optical flow systems will create patterns and illusions through their vision system in order to understand their environment. This can be a significant issue that has the effect of diminishing the object when the original image is processed by sampling to make a low resolution image. In addition, this system cannot distribute distance with the small size difference (In this experiment, the small size is under the 20cm). Therefore, we do not use HS directly, even though HS is similar to a biological sensory system. Instead, we use the LK algorithm.

2.3.2 Pyramid Lucas-Kanade algorithm

The LK algorithm is a local matching method that differs from the general optical flow algorithms. This implies that the LK algorithm can neglect the aperture problem that is often observed in optical flow through local approaches. Instead, the LK algorithm finds the nearest positions among the neighboring pixels, compared with a window of pixels. However, other assumptions such as brightness constancy, spatial coherence, and temporal persistence are acceptable to LK algorithm.

In the LK algorithm, the brightness constancy assumption can be applied to the equation as follows:

$$\frac{d}{dt}I(x,y,t) = \nabla I \cdot \mathbf{v} + \frac{\partial I}{\partial t} = 0 \tag{2.26}$$

where $\mathbf{v}$ indicates the optical flow vector and ∇I is a gradient intensity. To use a color image instead of a gray image, the differential intensity can be changed to gradient RGB values.

If ∇I is sufficiently large, we can rewrite the scalar equation into a vector equation by multiplying the transposed value of ∇I; that is, ∇I^t as follows:

$$\nabla I^t \nabla I \cdot \mathbf{v} = -\nabla I^t \nabla I_t \tag{2.27}$$

Next, we can intergate over the window W. Generally this model can be applied on the basis of the discrete pixels in image. So we can modify the integral to Σ operator as follows:

$$\iint \nabla I^t \nabla I \cdot \mathbf{v} dxdy = \iint \nabla I^t I_t dxdy \tag{2.28}$$

$$\sum\sum \nabla I^t \nabla I \cdot \mathbf{v} = \sum\sum \nabla I^t I_t \tag{2.29}$$

If the window size is small enough and spatial coherence is satisfied in the window, the $\mathbf{v}$ is constant in each point of the window W. So we can calculate $\mathbf{v}$ as follows (Barron et al., 1994):

$$\mathbf{v} = -(\nabla I^t \nabla I)^{-1} \nabla I^t I_t \tag{2.30}$$

However, this algorithm is incomplete when the movement or difference between two images arises from outside the range of the window. To overcome this problem, a new method, such as the pyramid LK, can be applied. The pyramid LK has *coarse-to-fine* approach according to the following steps (Bouguet, 2001). First, build a pyramid of images by smoothing and subsampling the original image. Second, select features at a coarse image and compute the optical flow. Finally, propagate the tracking features with computation of the optical flow at the next finer resolution. The pyramid LK algorithm can then be satisfied with global matching on the image.

Even though the model proposed in this paper usually uses background information instead of landmark information from the LK algorithm, it is not fully a background approach because the user of this algorithm can set the threshold values to choose the pixel of the image as a reference marker from among all of the pixels on each image. If a specific pixel on the image has values over and under the threshold, this pixel is discarded by the pyramid LK algorithm. In other words, the pyramid LK algorithm chooses reference markers on the image with fiducial value. Hence, this method automatically finds the marked and robust pixels as landmarks from the background through the pyramid LK algorithm. Therefore, the proposed model can be regarded as a combined landmark and background model.

2.3.3 KLT tracker

We chose the Kanade-Lucas-Tomasi (KLT) tracker (Barron et al., 1994) to express our homing vector because the KLT algorithm is faster than any other optical flow algorithms. According to Vardy and Moller (Vardy and Möller, 2005), the block matching algorithm (Jain and Jain, 1981) shows the best performance among the optical algorithms of differential flow methods. However, this algorithm is slow exponentially when the flow detection size is increased. On the other hand, the KLT algorithm has a pyramid approach to avoid outbursting of the computational time. Because the performance distinction between block matching and the KLT algorithm is small, even though block matching tends to dominate the precision of the homing vector, the decision to choose KLT is appropriate for a real-time experiment.

The KLT algorithm consists of two parts for making flow vectors. The first part is a feature extraction phase. In this phase, a corner detector can be chosen to find matched points on the reference image. One of the popular corner detectors was developed by Harris and Stephens (Harris and Stephens, 1988). To pick out better corners with this concept, Shi and Tomashi (Shi and Tomasi, 1994) suggested a threshold of eigenvalues on an autocorrelation matrix. An autocorrelation matrix derived from the Harris corner algorithm is as follows (Harris and Stephens, 1988):

$$A = \begin{pmatrix} \sum w_{i,j} I_x^2(x+i, y+j) & \sum w_{x,y} I_x(x+i, y+j) I_y(x+i, y+j) \\ \sum w_{i,j} I_x(x+i, y+j) I_y(x+i, y+j) & \sum w_{i,j} I_y^2(x+i, y+j) \end{pmatrix} \tag{2.31}$$

where I is intensity and $w_{i,j}$ means the weight that can be chosen as a general Gaussian model. If the smaller of two eigenvalues in the autocorrelation matrix 2.31 is bigger than a certain threshold λ, this feature can be regarded as a good corner. In this process, a number of corners, as many as can cover the image uniformly, are chosen based on descending order of the eigenvalues. Uniform distribution of the corners on the image can help to delineate the minute flow patterns for FOC and FOE. This idea is also applied to the KLT tracker (Barron et al., 1994) to measure the differential flow between the two images.

After detection of the corners, the second phase is activated. In the second phase, the pyramid LucasKanade (pyramid LK) algorithm is used from the corners. Compared with previous methods that use the automatic corner detection algorithm (Vardy and Oppacher, 2003, 2004), the proposed method has the advantage of performance for returning to home due to processing of the LK algorithm. Basically, the optical flow above two dimensional cases cannot be calculated due to the reduction of the equations, so additional assumptions are needed to create optical flow. Brightness constancy is one of the important assumptions in the optical flow algorithm, which means that a set of pixels in the same area is not changed with a small movement of the image.

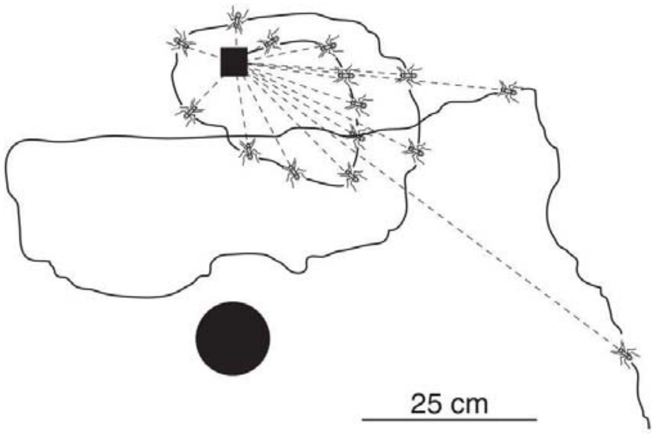


Figure 2.14: Trace of the learning walk of Namibian desert ant *Ocymyrmex* when black landmark is suddenly emerged(Black circle) in front of the nest (Reprinted from Graham et al. (2010).

2.4 Probabilistic approach to local visual navigation

2.4.1 Overview

According to Graham et al.(Graham et al., 2010), the desert ant uses several nest-oriented snapshots close to the nest to detect its home direction from distant release sites. Surprisingly, if the ant encounters an unfamiliar environment from the nest, the ant slowly circles the nest entrance in a spike-like way and aligns own body to the nest. However, after acclimatizing to its surroundings, the ant does no longer moves rotationally (Müller and Wehner, 2010). This tendency of the ant connotes that it can facilitate route learning by using vision information with active sensing. Figure 2.14 shows an example of the ant's movement track when environment around nest is changed. From this biological background, several measurement and control strategies can be set to develop efficient navigation of the agent.

The snapshot hypothesis is one of the main ideas used to describe the sensory system of an insect (Wehner and Räber, 1979; Cartwright and Collett, 1983). This hypothesis holds that an insect can only store two images, taken from the current position and a reference position, to estimate a homing direction. The home vector can be derived by comparing the difference between two images and the insect can then return to its home. In the snapshot model, the components of the image vector can be made from only contained intensity values without depth information between the agent and objects. This implies that local visual homing using intensity information can find a homing vector because a method that extracts depth information from the original image needs to have high complexity and many sensory systems and results since a measurement model usually has high dimensionality (Stürzl and Zeil, 2007). Another characteristic of the snapshot hypothesis is the use of the raw image directly without any preprocessing. This idea also helps to save computation time compared with other methods that are accompanied by complex image processing.

In biology, active sensing is also one of the fundamental factors that demonstrate an organic system. This idea can be applied to the control model of agent localization. When a specific situation has many uncertainties, active sensing can reduce the possibility of faulty estimation of the state and guarantee to idempotent of correct estimation for system. For reliable active sensing, compass information is needed to set the reference direction of the agent. Generally, an insect can use celestial compass information for orientation and alignment because sky information has robustness of information from assorted noise (Homberg, 2004). Therefore, in this paper, the agent can use compass information from external or internal factors to correct the alignment and to help maximize the expectation for agent localization to weight the system dynamics of the Bayes filter.

Although mammals and insects are different species, the concepts of place cells and grid cells using mammals to recognize a map are important bio-inspirations in the design of grid-based approaches using discrete belief (McNaughton et al., 2006). However, since place cells have a spatial limitation for expressing a whole map, a model imitating pure place cells is not suitable for describing long range navigation. In real life, mammals cannot correspond to place cells directly on the long range space. Instead, they can use place cells recursively, by matching various patterns stored in grid cells. For example, if the environment around the agent is changed but the agent does not move, even though the place cells still activate at the same range, the firing range of the grid cells can be changed. This means that mammals can describe a broad map in spite of limitations of the number of place cells. Likewise, one of limitations for this model is also the confined observable range for the reference image. If the agent leaves the observable range, descriptors of all positions become inconsistent because of a lack of information. To relieve this limitation, several images are stored as reference information, like grid cells. In other words, scattered snapshot images can help with localization for long range navigation. From these basic ideas, we can develop an agent localization model based on an active sensing strategy with a bio-inspired measurement model.

2.4.2 Bayesian approach to homing navigation

In this paper, we chose a differential flow method that uses extraction of features and matching in correspondence methods based on optical flow algorithms (Vardy and Möller, 2005) among the many approaches available for local visual homing navigation. The advantages of differential flow methods are their moderate performance and fast processing speed. To reinforce this method, we also apply the average landmark vector (ALV) method (Lambrinos et al., 2000) to a differential flow method. The optical flow algorithm from the differential flow method creates the flow vector, which is a deviation between two images taken from a reference position and a certain position. The start point and end point for each flow vector result in angular values of the unit vectors at different positions. From this information, the average landmark vector can be calculated by summing all unit vectors. Consequently, the homing direction is derived by subtracting the two vectors and the agent can move with direction to arrive at the home location using only visual cues. The combination of ALV and the differential flow method can reflect the distribution for the focus of contraction and focus of expansion between the two images (Vardy and Möller, 2005) as well as the difference for each landmark. This is unlike the ALV algorithm because the ALV algorithm can estimate the homing vector by calculating the angular position difference of several landmarks, but the corners made from the differential flow algorithm are not constant for the whole map. The use of only the fcolor cues of landmarks can result in the occlusion problem, which degrades the performance of homing navigation (Yu and Kim, 2011a,b, 2012). Figure 2.8 introduces the FOC and FOE distribution for panorama images.

Nevertheless, to avoid increasing the computational time problem, the probabilistic method is rarely applied to bio-inspired homing navigation, even though a Bayesian approach can help to reduce the error arising from a noisy environment. To maintain efficient computation, we propose only a one dimensional Kalman filter for choosing the homing direction. The input data of the Kalman filter is produced from the measurement model that calculates the homing vector. We assume that the measurement data have uncertainty as a Gaussian distribution. Generally, the distribution of homing vectors on the map has a specific and consistent directivity to home. Therefore, by accumulating homing vectors, the Kalman filter can estimate the homing vector at the next position.

One drawback of the Kalman filter is that the filtering of results is difficult to change when the Kalman filter finds a major factor from the measurement model. Nevertheless, the measurement model of the ALV-based KLT is usually constant except for the specific case where the agent encounters external noise. Using this character of the measurement model, the agent can set a homing direction in a noisy environment.

2.4.3 Bayesian filter

Bayesian filtering is a popular method used in probabilistic robotics (Thrun et al., 2005b). It can serve to reduce uncertainty of the model with an iteration method. In probabilistic robotics, Bayesian filtering can be used to resolve problems such as automatic localization and mapping. Basically, the agent does not know its own initial state, but through iterative methods with Bayesian filter, the agent can know its own state to estimate the expectation maximization of $Bel(x_t)$.

$Bel(x_t)$ means the state of the agent after measurement update z_t. To get a measurement update, the agent can calculate $\overline{Bel(x_t)}$ as a control update with movement u_t. From this idea, we can derive $Bel(x_t)$ and $\overline{Bel(x_t)}$ as the following:

$$Bel(x_t) = P(x_t|z_{1:t}, u_{1:t}) \tag{2.32}$$

$$\overline{Bel(x_t)} = P(x_t|z_{1:t-1}, u_{1:t}) \tag{2.33}$$

where x_t is the state of the agent when at time is t. Due to the uncertainty of this state, x_t can be described as a probability of the state's belief. To converge the two concepts with filtering, a general model of the Bayesian filter can be described as the following:

$$\overline{Bel(x_t)} = \int P(x_t|u_t, x_{t-1})Bel(x_{t-1})dx_{t-1} \tag{2.34}$$

$$Bel(x_t) = \eta P(z_t|x_t)\overline{Bel(x_t)} \tag{2.35}$$

where equation 2.34 is a control update and equation 2.35 is a measurement update.

Unfortunately, even though the Bayesian filter is powerful enough to converge in a noisy environment, the computational load of a Bayesian filter is very high. This means that a simple filter approximated from an original Bayesian filter is required to apply to real cases. In the proposed model, we use a Kalman filter and a non-parametric filter such as a histogram filter and particle filter to estimate the agent's state. To reduce inefficient computation, we design the measurement model as only one-dimensional information. Details of this probabilistic model will be described in section 5.

2.4.4 Grid-based approaches

The histogram filter as a grid-based approach is one type of discrete Bayes filter (Thrun et al., 2005a).This filter is similar to the place cells in the mammalian brain. Although the histogram filter has a critical drawback in that it requires a high computation load and memory to keep the spatial information and to update every observation (Fox et al., 1998, 2003), this filter can be a good choice due to the particularity of the measurement model. The main particularity is the range limitation for the reference image. Each omni-directional image taken from a reference position cannot contain objects connected to the next image taken from current position when agent moves out of a specific range. In this case, the KLT algorithm cannot detect movement sizes of objects between two images. This means that broad map for inefficiently expressing with histogram filter is needless. In the next section, details about the handling methods for this problem will be introduced, but particularity does not imply that other filtering methods, like the Kalman filter or the particle filter, are not useful in this case. We merely intend to show that a system with a histogram filter also can show performance convergence for localization with low computation and memory and we want to describe the design of place cells as similarly as possible. If the user wishes to design using a different Bayes filter, this can also be a reasonable system with specific tuning for several parameters.

For the discrete Bayes filter algorithm, predictive belief with the control update and belief after the measurement update can be described as the following::

$$\overline{Bel(x_t)} = \sum_i P(x_t|u_t, x_{t-1}) Bel(x_{t-1}) \tag{2.36}$$

$$Bel(x_t) = \eta P(z_t|x_t) \overline{Bel(x_t)} \tag{2.37}$$

where $\overline{Bel(x_t)}$ and $Bel(x_t)$ are predictive belief and belief, respectively. η is a normalizer. x_t is a spatial state of where agent is. Estimation of state x_t is dependent on the most recent control u_t in the control update. Basically, previous cases for $t-1$ are not considered because the Bayes filter obeys the rule as a Markov assumption or a complete state assumption. z_t is a measurement of the agent with perceptual sensors. If the agent has a visual sensory system, z_t is image information from an omni-directional camera.

To improve the performance of Bayes filter, measurement z_t can be simplified as $\Delta\theta_t$ through the homing vector of a local visual homing navigation with intensity information. The agent can be predicted to angle for each relative position from the reference position without the experience that the agent visits the position and measures the homing vector using a descriptor because two-dimensional distribution of the homing vector around the reference position consists of centralized vectors for each place. Figure 5.3 displays an example of a predicted vector map around the reference position. With the vector map information, the agent can calculate a difference angle between the predicted case and the real case and assign weighting points for each case in a grid map. Low computation is needed because the measurement model contains only one-dimensional information to describe angle differences. Details of the measurement model will be introduced in subsection 2.2.

Active sensing strategy of the agent can help with fast convergence to the correct position with lower error. The agent can recognize the reference angle from a visual compass algorithm. This means that the agent can decide its own direction of movement without information about position and orientation when the agent tries to actively sense with its own specific action. Since a visual compass is not always correct, noise measurement can interfere with the original descriptor to find the alignment angle, so uncertainty of alignment is also considered. The agent can, at least, be sure of its direction of movement with high probability. This concept can be applied to control update $P(x_t|u_t, x_{t-1})$. Subsection 2.3 provides a formal discussion of this active sensing strategy.

2.5 Overview of proposed model

Local visual homing methods with intensity information are based on the bio-inspired navigation. Like an insect visual mechanism, this methods also use snapshot hypothesis. The main criteria that seperates out different methods is a methodology that measures discrepancy between two images taken from reference and current positions in snapshot hypothesis. Generally, optical flow can be regarded as visual clue that can be described discrepancy.

Among the many approaches for local visual homing navigation, we choose differential flow methods that use extraction of features and matching in correspondence methods based on optical flow algorithms. The advantages of differential flow methods are moderate performance and fast speed to processing. Even though block matching algorithm shows best performance among the optical algorithms of differential flow methods, Kanade-Lucas-Tomashi (KLT) algorithm (Barron et al., 1994) is chosen to express the homing vector at the each position far from the reference position because KLT algorithm is faster than block matching algorithm generally. From KLT algorithm, the homing vector can be estimated by just adding all the vectors that indicate difference between two images. To track the homing vector, the agent can be returned to home. To simplify calculation, the average landmark vector can be proposed.

However, the research about localization algorithm with local visual homing methods based on bio-inspired mechanism has been made slow progress compared with the reserach of homing navigation so far. Commonly, insects have regarded that they cannot recognize own position because the place cells in their brain are not existed. To reach the homing position, insects are considered to using only directional information from internal and external clues. But, apart from the biological results, we design a vision system and a strategy of the insect with probabilistic approaches to help to make a efficient localization system for the engineering. For instance, the angle of result vector from the local visual homing method can be considered as a good descriptor that can be used to distinct the state of agent for the meaurement model with low dimensionality instead of the homing vector. This feature can reduce a complexity of computation for probabilistic approaches based on Bayes filter.

The specific phenomenon that the set of homing vectors on the map has convergence center-biased specific location also helps to predict patterns of angle around reference positions. Contrary to general localization methods, pre-searching to recognize specific descriptors on each unvisited position is unnecessary in this method using vector convergence. Instead, as a vicarious factor, an omni-directional image taken from reference position is needed. This idea is from assumption that an image taken from location near home position varies smoothly and monotonically with spatial distance compared with image taken from reference position. As a result, the agent with image taken from reference position can find their positional information as well as the homing direction through Bayes filter.

2.6 Summary of Chapter 2

In this chapter, several background ideas are introduced. First, local visual navigation can be distributed using two concepts such as holistic methods and corresponding methods. We use a differential flow method in corresponding methods to estimate the homing direction because of its robustness for a complex background and its fast speed for computation. To design the differential flow method, a second KLT tracker, based on the LK algorithm, is introduced for optical flow. The KLT has two processing parts, including corner detection and a pyramid LK algorithm. Through these algorithms, the KLT can find corners and matching points between two images.

Third, probabilistic approaches are used to reduce the uncertainty of the environment with external noise. However, the Bayesian model is too complex to apply directly to a real case, so we consider a Kalman filter and a non-parametric filter to reduce the computational load. In the proposed model, input can be simplified as only one-dimensional information.

Chapter 3

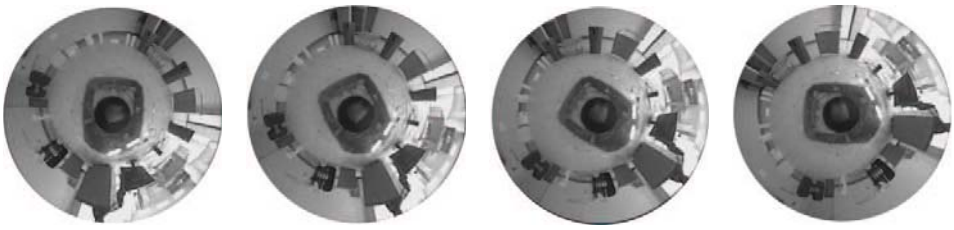
Accumulative optical flow navigation

In this chapter, we suggest a navigation method that uses a pyramid LK algorithm invented by Lucas and Kanade (Lucas and Kanade, 1981). The LK algorithm is distinctive because it is a pyramid process related with a *coarse-to-fine* approach; it can make long flow vectors with windows of confined sizes by iteratively processing a hierarchical pyramid with low resolution pixels to high resolution pixels. From this, we can acquire the pattern of movement vectors for agent locomotion and obtain distance and direction information by calculating vector information. Compared with existing algorithms that use KLT algorithms (Vardy and Möller, 2005), the proposed algorithm in this paper uses both the original omni-directional image and a panoramic image. From the omni-directional image, we can easily get information about the focus of expansion (FOE) and focus of contraction (FOC). The contents of this chapter were published in a conference proceedings (Cha and Kim, 2012b) and a journal (Cha and Kim, 2012a, 2013a).

3.1 Application of visual navigation with an optical flow algorithm

3.1.1 Aligned angle of the agent

Generally, saccade can straighten a distorted angle by using rotational components of optical flow vectors to estimate distance (Lindemann et al., 2005). We used the



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

Figure 3.1: Image rotation processing to find specific direction at the reference position through minimizing rotational vectors: (a) original image, (b) rotating 30 degrees, (c) rotating 60 degrees, (d) rotating 90 degrees

saccade of insects, to test the workings of an aligned agent method. If the distorted angle adds onto the original angle, the snapshot images make additional rotational vectors. Comparison of the original image and a new image that was a distorted angle produces rotational vectors whose lengths depend on the movement. Hence, we can find the original aligned angle at the point of minimum size of the vector by rotating agents with the LK algorithm. That is, for two given snapshot images, we can rotate a snapshot image to minimize the sum of rotational vectors. The rotational angle then determines the alignment angle to return home.

Figure 3.1 shows the image processing of rotating with automatic interpolation. We can find the same angle as that taken from home by calculating the optical flow. If the picture has distorted parts from the rotating movement of the agent, additional rotational vectors add to existing flow vectors. This means that a suitable alignment angle can be found at the position by obtaining the minimum size of the total vectors. Figure 3.2 shows an example of the process of finding the angle with the minimum vector size of optical flow.

The total strategy is shown in Figure 3.3. First, the agent captures the image at home. Second, upon arrival at the goal position, the agent also captures another image. By comparing the two images, the algorithm finds the error from the rotational factor of the optical flows. Third, to minimize the rotational optical flows, the agent directly rotates itself or the captured image at the goal position. Through processing to calculate the total length of the optical flow vectors, we can arrange the agent to achieve the same angle at the home position.

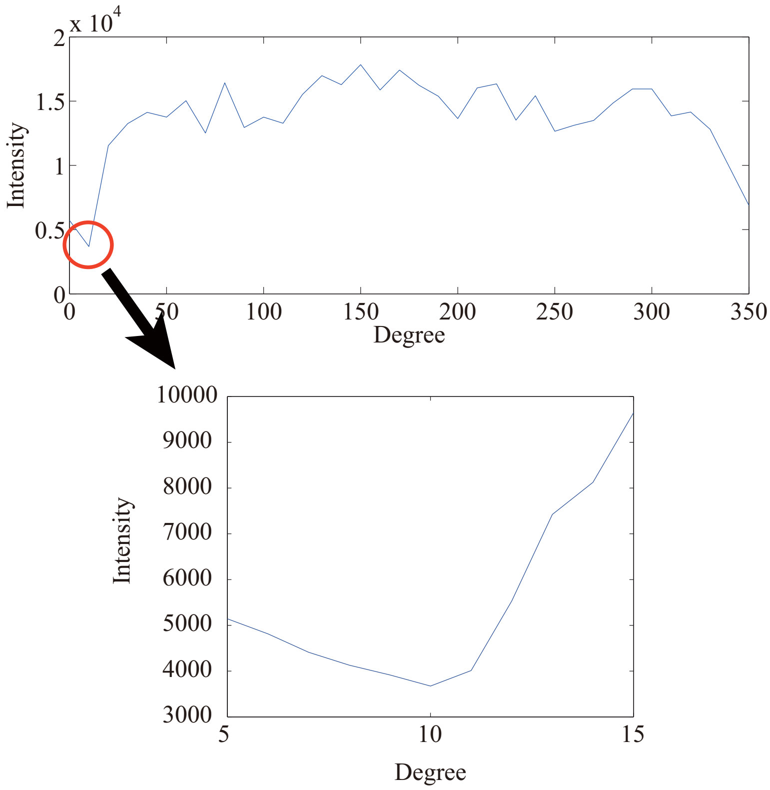


Figure 3.2: Distribution of intensity of optical flow with two images: 0 to 350 degrees changes per 10 degrees. Initially, matching angle is 10 degrees(red circle).

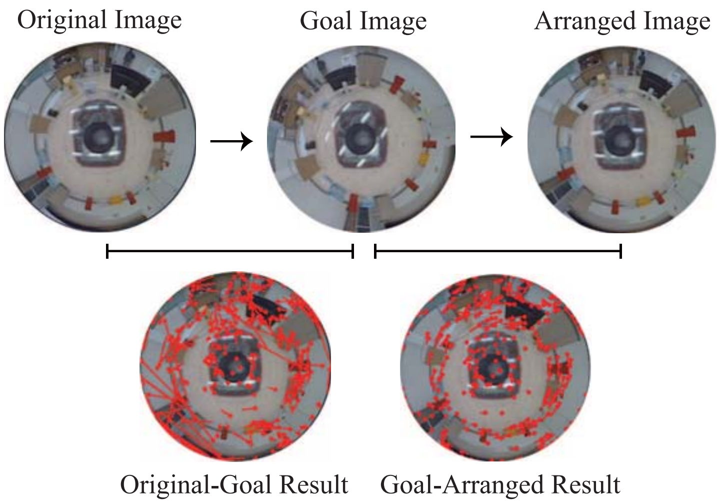


Figure 3.3: Strategy of image matching based on LK optical flow algorithms.

3.1.2 Estimation of the agent state

We introduce the two methods - path integration with a distance and angle map and path integration with a movement classes - as follows:

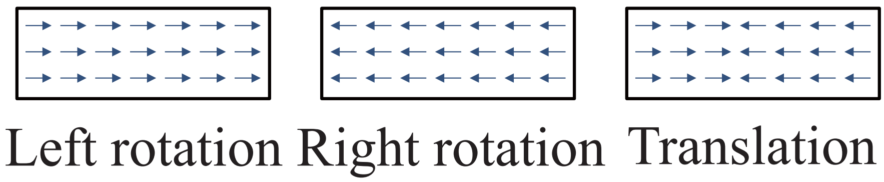


Figure 3.4: Several optical flow vector patterns for discretizing class method

Path integration with a distance and angle map The proposed method uses optical flow vectors on the camera and snapshot images to estimate the states of an agent such as direction and moving distance. Vardy and Oppacher (Vardy and Oppacher, 2003) proposed a method similar to ours to find the FOE and FOC to estimate the direction of agents. We find the FOE and FOC points calculating just the sum of vectors with the omni-camera as follows:

$$\vec{d} = \Sigma \vec{v}_i \tag{3.1}$$

where $\vec{d}$ is the sum of vectors and $\vec{v}_i$ is each optical flow vector. We calculated the optical flows in the omni-directional image, not a panoramic image in this case. We also use the panoramic image for the later cases. The states of vector movement are preserved on the omni-images without any other distortions. So, we can reduce computation time for transforming the images.

Moving distance estimation for the agents can be calculated from the length of total vectors. The size of the vectors is proportional to the distance of the objects and the speed of the agents. Here, we assume that objects are far from the agent, and that the moving distance is approximated as the length of the optical flow vector.

Path integration with a movement classes Unfortunately, the previous model cannot be applied to distorted environments that have heterogeneous placement of the objects. For instance, if the distance from the agent and objects is changed, the length of the optical flow vector can also be changed. As a result, threshold values decided from initial conditions are useless for different environments. We must reduce patterns of optical flow to prevent incorrect movement or recognition of an agent for a changed environment.

In this case, we divide the patterns into three cases: straight ahead, left rotation and right rotation of the agent. If the optical flow vectors on the image have the same direction, the agent can determine its own direction as left or right because of the specific tendency for objects in the environment to move in the same relative direction when the agent rotates with a direction of left or right. If the optical flow vectors on the image have a different direction, the movement of the agent can be regarded as a translational movement. The classified criterion is the summing of the left and right optical flow vectors and comparing with certain threshold values. In this manner, the agent can estimate its own movement easily around the heterogeneous placement of the objects.

We extend the detectable range by using the panoramic image transformed from the omni-directional image. The three patterns are shown in Figure 3.4. The agent can calculate its present spatial state by accumulating optical flow vectors. We enhance performance of estimation by assuming that unit movements of the agent for rotational and translational classes are constant. In the real world, insects are believed to refer to their own classes to apply an internal encoder their estimate spatial states.

The agent uses Cartesian coordination transformed from polar coordination for the spatial state to describe biological fidelity. A trigonometric function can help to calculate the agent's position. The x and y position can be designed using a triginometric function with the angle and the distance of the agent. The accumulation phase will be activated from the each of the classes by summing the current position and the head direction.

This method is similar to the summing method but differs in the classified criteria which are obtaining by counting the left and right optical flow vectors and comparing them with certain threshold values. The computational time is smaller than the summing method, but this method has a limitation as it cannot be used for cases with many classes.

3.1.3 Experimental procedure

We tested our optical flow experiments in an office environment. This arena is surrounded by a complex background and consists of 2-dimensional squares about 1 meter wide and 2 meters high. Figure 3.5 shows a mobile robot and the arena image. A Pioneer robot acts as the agent ans uses the LK optical flow algorithm to return home after reaching a goal position.

(a) (b)

Figure 3.5: Image of arena: Arena made in general laboratory environment. (a) is pioneer robot as agent with mini computer which calculates the optical flow vectors. (b) expresses surroundings of arena. Red range on image indicates the area where the omni-directional camera of agent can't see.

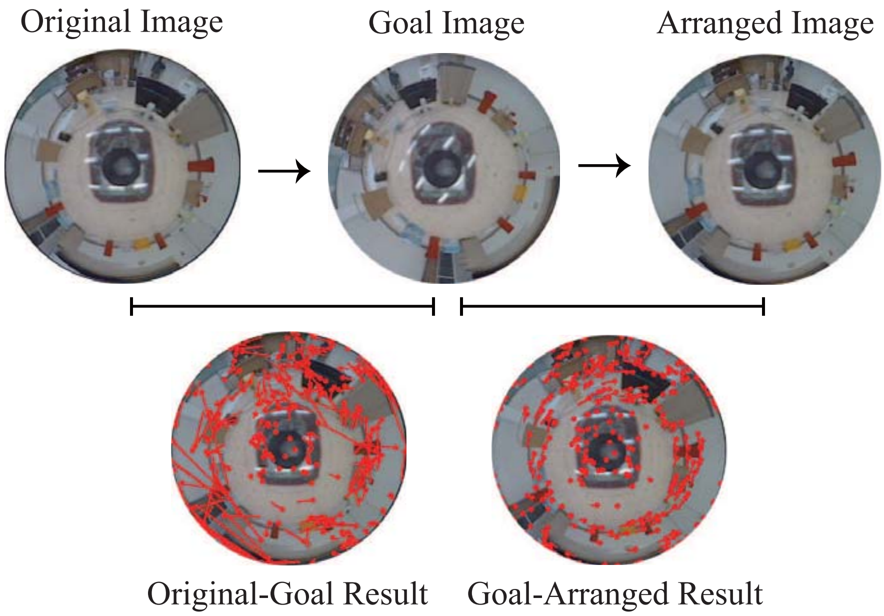


Figure 3.6: The strategy of image matching based on LK optical flow algorithms.

3.2 Path integration with a distance and angle map: equal-distance landmarks

We first test the performance of the optical flow algorithm, by designing a specific area surrounded with diverse objects in the lab. Allocated objects mean factors that can make flow vectors on an image to estimate the spatial state of the agent. In this arena, we check the agent performance of the rate of returning to home. Precise results are derived by setting the allocated objects in the environment at the same distance away from the initial position of agent in order to satisfy the equal distance assumption.

Case1				Case2			
N	L(cm)	E(°)	E(cm)	N	L(cm)	E(°)	E(cm)
1	160	3.48	4.31	1	160	-4.02	-15.1
2	160	-3.35	-6.84	2	160	3.24	-7.54
3	160	1.03	-1.29	3	160	1.94	5.60
4	160	-0.10	0.43	4	160	-3.77	-10.3
5	160	2.46	-6.12	5	160	-1.23	4.18
Avg.		2.08	3.80	Avg.		2.84	8.54
Std.		1.48	2.85	Std.		1.21	4.32

Table 3.1: Degree error and distance error for this case with image matching method based on LK algorithm: N is the total number experiments. L is total length for navigation. E means calculation error for homing route. Avg. and Std. means average and standard deviation, respectively.

Figure 3.6 shows a method whereby the agent can estimate a homing direction from optical flow vectors. In this way, the agent can accumulate spatial information and find a homing direction. Table 3.1 shows the results of cases 1 and 2. The route in case 1 is designed as a zigzag line and the route in case 2 is designed as a random line. After the agent is moved in a specific line, it can return home by calculating its movement through estimations of accumulative patterns of optical flow. Both cases of navigation require moderate performance to estimate path integration with optical flow.

However, this case is a nearly ideal model. These experimental results cannot guarantee that this accumulation of optical flow patterns will have robust performance for a real case, although they can confirm the performance of visual descriptors made from optical flow vectors. Therefore, we carried out additional experiments with a general environment in a lab. Detailed results of these general cases are presented in the next section.

(a) (b) (c)

Figure 3.7: Example using two images for calculation of optical flow vectors: (a) Image taken at the home position, (b) Image taken at the aim position, (c) Result of LK algorithm

3.3 Path integration with a distance and angle map: realistic environment

3.3.1 Performance

We set the route on the arena for movement of a Pioneer robot as an agent. We tested the route five times to estimate the performance of image matching based on the LK algorithm. Figure 3.7 shows the pictures of the start and end points of the exploration and the result of the LK algorithm with these pictures. In a real case with a complex background, the patterns in the omni-directional camera consist of irregular vectors. By average processing of all vectors, we can choose a representative vector for the results of the LK algorithm.

Table 3.2 presents the results of angular error and distance error for navigation using image matching based on the LK algorithm. This algorithm gives a solution directly to the agent by estimating a spatial state by calculating flow vectors. The routes of the agent in the arena are shown in Figure 3.8.

In some cases in the arena, the results of performance for each case are not constant because external noise (such as light, nonuniform background, and the state of camera) interferes with the process. Light is one of main problems that determine the performance of image matching based on the LK algorithm. When the light changes frequently, the vectors cannot be created consistently for changing at the unexpected points of the image. The LK algorithm can make vectors for alternate points of the image. A nonuniform background reduces the flow vectors by removing productive el-

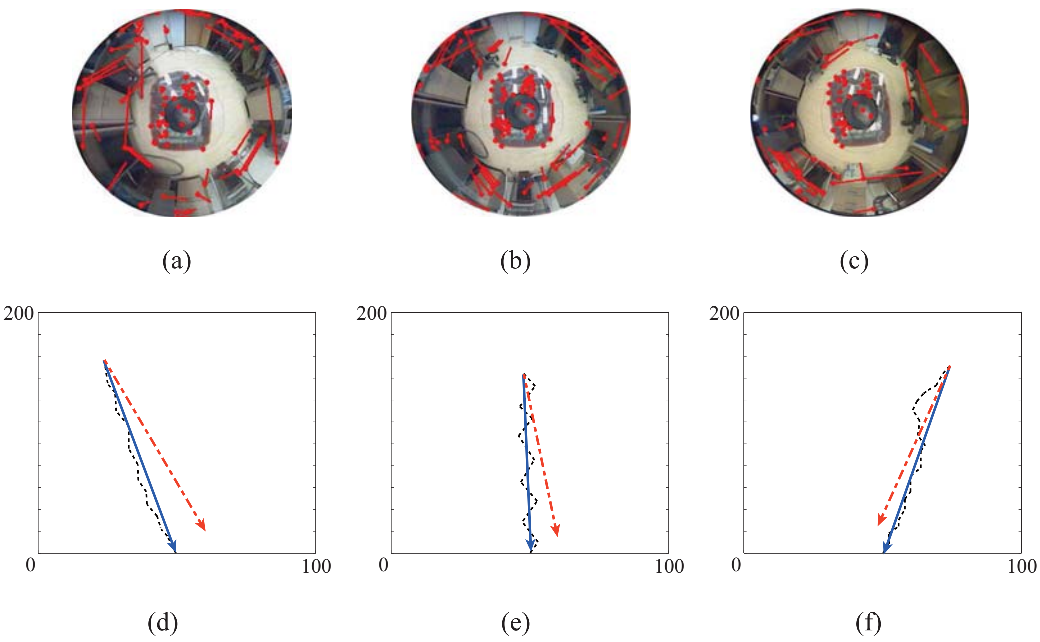


Figure 3.8: Three cases of homing navigation based on optical flow using image matching approach: (a) and (d) are results when moving left. (b) and (e) are results when moving straight forward. (c) and (f) are results when moving right. Black dotted line is the route navigation of agent, blue arrow indicates desired movement of agent and 5 red arrows indicates teh actual movement of the agent. Agent movement is executed in 200cm x 100cm arena.

ements in the image. If the state of the camera is unstable, the vectors can be distorted by external noise. For instance, vibration of the camera from a fraction of the floor can distort the optical flow result.

We also tested an accumulative model in a wide range laboratory map chip. Unlike the previous cases, this map has a complex background that is far from the agent. This factor creates uncertainty in the accumulative navigation because the patterns of the flow vector can be distorted by complex landmarks. These data will be used in chapter 5 to evaluate the performance of probabilistic filtering.

Table 3.3 shows the results for the wide environment in the laboratory. This experiment was executed a total of 10 times. In these cases, even though noise is increased by the complex environment, the agent still knows the start position of movement with moderate error using only visual information.

Case1				Case2				Case3			
N	L(cm)	E(°)	E(cm)	N	L(cm)	E(°)	E(cm)	N	L(cm)	E(°)	E(cm)
1	160	16.4	11.5	1	160	14.5	18.0	1	160	12.3	15.5
2	160	17.2	26.1	2	160	10.3	23.5	2	160	16.3	15.5
3	160	8.9	4.5	3	160	15.5	16.2	3	160	18.7	13.2
4	160	14.2	21.1	4	160	10.8	9.4	4	160	16.2	24.5
5	160	17.6	32.7	5	160	13.1	11.5	5	160	13.5	20.2
Avg.		14.9	19.2	Avg.		12.8	15.7	Avg.		15.4	17.8
Std.		3.6	11.3	Std.		2.3	5.7	Std.		2.5	4.5

Table 3.2: Degree error and distance error for this case with image matching method based on LK algorithm: N is the total number experiments. L is total length for navigation. E means calculation error for homing route. Avg. and Std. means average and standard deviation, respectively.

Wide case					
N	L(cm)	E(°)	N	L(cm)	E(°)
1	200	-16.05	6	200	-28.66
2	200	-23.50	7	200	-21.21
3	200	-25.22	8	200	-22.36
4	200	-32.10	9	200	-26.94
5	200	-42.99	10	200	-9.75
Avg.					-24.78
Std.					8.99

Table 3.3: Degree error and distance error for this case with image matching method based on LK algorithm: N is the total number experiments. L is total length for navigation. E means calculation error for homing route. Avg. and Std. means average and standard deviation, respectively.

3.3.2 Robustness of low resolution

Another issue that arises when extracting optical flows from several images pertains to the quality of images. High quality images do not always guarantee good performance of optical flow vectors. In nature, many insects use optical flow to control themselves with their low resolution eyes (Franceschini, 2004; Nordström et al., 2006). but the

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

Figure 3.9: Sampling original images to check robustness of low resolution for image matching method based on LK algorithm: (a) 6 sampling, (b) 12 sampling, (c) 18 sampling, (d) 24 sampling

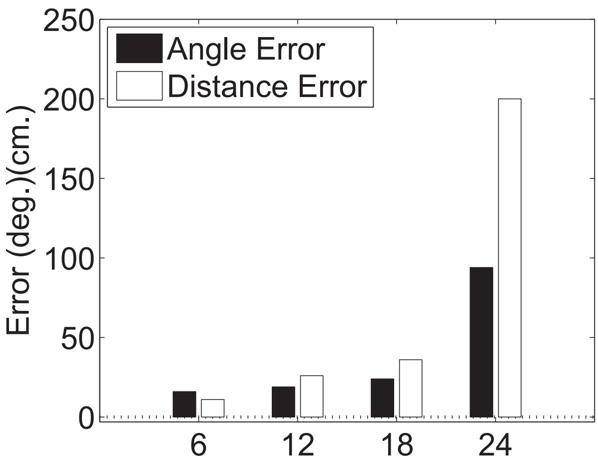


Figure 3.10: Error rate for each number of sampling

consistency of quality is a very important factor with image processing of optical flow since differences in quality causes confusion in pixel matching and difficulty in finding a similar nearby color range.

We tested the robustness of low resolution for the proposed algorithm. Initially, a low resolution image can be built by sampling the original image, as shown in Figure 3.9. In a low resolution image, even distinguishing the object is more difficult than in the original image, and the element of flow vectors can stand out in the image.

Figure 3.10 shows that even low resolution images can effectively find the homing vector. Results of the performance are good, except for an extreme case of 24 samplings that shows bad performance. The optical flow vectors and their integration can successfully guide the agent to return home. From this result, this measurement model can be regarded as a robust system for low resolution images.

Figure 3.10 therefore indicates that a progressively lower resolution image does not degrade the performance of the homing navigation with image matching based implicitly on the LK algorithm.

On occasion, better performance is obtained for cases with a low resolution state of the image provide better performance than with a high resolution case. However, over the threshold of resolution, the performance of homing navigation is not guaranteed to result in an arrival at the home location. As shown in Figure 3.10, 24 of the samplings cause the agent to fail while traveling on the map with an optical flow algorithm. The original image sizes are about 350 x 350, but this robustness of resolution cannot be applied to a method that directly uses a diminished image from sampling. The image must be resized to the size of the original image because the sampled image cannot consider movement of the agent and the distance between specific pixels on the image. Thus, if the system uses a sampled image directly, the performance rapidly degrades.

3.4 Path integration with movement classes

We can test three classes and five classes to make a robust decision of the homing direction in this paper. Figure 3.4 indicates the example of the three classes. If one of the optical flow patterns is activated after movement of the agent, the agent can estimate its own movement with the results of the patterns. In the path integration, the agent can accumulate spatial information as a trigonometric structure about its own position and head direction of its own angle through interpretation of the results of the given patterns. The method can be derived as this equation as following equations:

$$f(\vec{d}) = \begin{cases} \text{Left rotating movement} & \text{if} \mid \vec{d}_L \mid - \mid \vec{d}_R \mid > \varepsilon \\ \text{Right rotating movement} & \text{if} \mid \vec{d}_L \mid - \mid \vec{d}_R \mid < -\varepsilon \\ \text{Straight forward movement} & \text{if} -\varepsilon < \mid \vec{d}_L \mid - \mid \vec{d}_R \mid < \varepsilon \\ \text{No decision} & \text{if optical flow vectors do not exist.} \end{cases} \tag{3.2}$$

where $\vec{d}_L$ and $\vec{d}_R$, respectively, are the summation of the length of the vectors that indicate the left direction and summation of the length of the vectors that indicate the right direction. ε is a non-negative value used as a threshold to determine the homing direction after the path integration phase of the agent. If the assumptions in the KLT algorithm are violated by external factors, the optical flow vectors cannot be made. In this case, the agent cannot decide its own movement.

This method can be also applied to extended number classes; for example, the method with five classes can be made as follows:

$$f(\vec{d}) = \begin{cases} \text{Left large rotating movement} & \text{if} \mid \vec{d}_L \mid - \mid \vec{d}_R \mid > \varepsilon_2 \\ \text{Left small rotating movement} & \text{if} \varepsilon_1 < \mid \vec{d}_L \mid - \mid \vec{d}_R \mid < \varepsilon_2 \\ \text{Right large rotating movement} & \text{if} \mid \vec{d}_L \mid - \mid \vec{d}_R \mid < -\varepsilon_2 \\ \text{Right small rotating movement} & \text{if} -\varepsilon_2 < \mid \vec{d}_L \mid - \mid \vec{d}_R \mid < -\varepsilon_1 \\ \text{Straight forward movement} & \text{if} -\varepsilon_1 < \mid \vec{d}_L \mid - \mid \vec{d}_R \mid < \varepsilon_1 \\ \text{No decision} & \text{if optical flow vectors do not exist.} \end{cases} \tag{3.3}$$

where ε_1 and ε_2 are non-negative values used as thresholds to determine the homing direction after the path integration phase of the agent.

n this paper, the left rotating movement in the three class method and the left small rotating movement in the five class method have the angular size of the 5 degrees, the right rotating movement in the three class method and the right small rotating movement in the five class method both have an angular size of the 5 degrees. The size of the large rotating movement in both class sets is 10 degrees. The movement size for straight ahead forward in both classes is 30 centimeters. We assume that the agent has confined movement patterns as seen with insects. Initially, the insects have nearly the same size of movement because the size of the insects is generally very small. Consequently, the agent must choose one of the patterns among three or five cases.

Instead of the summing method for optical flow vectors, the counting method can be regarded as the alternative choice. The counting method considers the numbers of corners that indicate left or right directions. This model can be derived as follows:

$$f(\vec{v}) = \begin{cases} \text{Left rotating movement} & \text{if} \quad n(\vec{v}_L) - n(\vec{v}_R) > \varepsilon \\ \text{Right rotating movement} & \text{if} \quad n(\vec{v}_L) - n(\vec{v}_R) < -\varepsilon \\ \text{Straight forward movement} & \text{if} -\varepsilon < n(\vec{v}_L) - n(\vec{v}_R) < \varepsilon \\ \text{No decision} & \text{if optical flow vectors do not exist.} \end{cases} \tag{3.4}$$

where $n(\vec{v}_L)$ and $n(\vec{v}_R)$ are the number of vectors that indicates the left direction and the number of vectors that indicates the right direction, respectively and ε is the threshold value used in the counting method to distinguish the patterns of the intended movement. This method is simpler than the summing method, but it cannot be applied to the cases with a number of the classes above four, since the weight of the vectors is always the same. We tested this method on the three class case and compared it with the summing method.

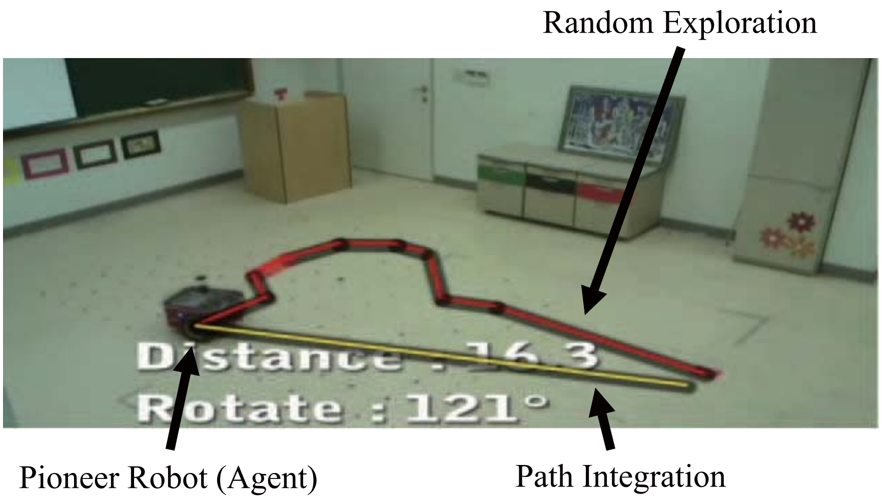
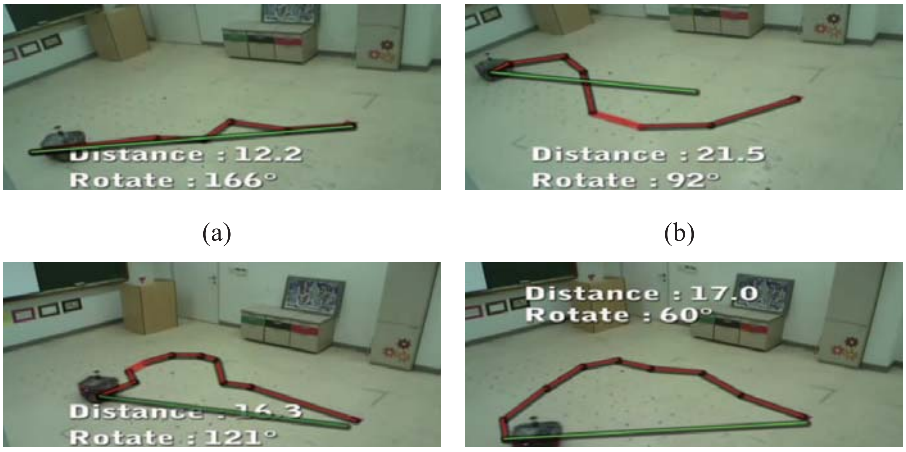


Figure 3.11: The experiment of path integration: There are four phases which are required for path integration with optical flow such as initial phase, wandering phase, path integration phase, and return phase.

All phases are introduced in Figure 3.11. This experiment was set in a general laboratory with heterogeneous objects. In the initial phase, the agent initializes its internal state and any other sensory inputs. The agent can receive visual information from the omni-directional camera with the LK algorithm and classify the patterns of optical flow vector in a wandering phase. Classified visual data can be converted to a spatial state with a trigonometric function and accumulated using previous information until the path integration phase. In the path integration phase, the agent stops its wandering movement and estimates its own position with the accumulated state. The final angle and position calculated from the optical flow vectors are needed by the agent to find a way to return. The agent can be returned to the home position based on the results of the calculation in the return phase. When the number of patterns is five, the agent can also find its home position with the optical flow vector in the laboratory case. However, the result is incorrect when compared with the case with three patterns because this system with the five patterns contains criteria measured for length of the optical flow vector. The length of the optical flow vector can be affected by the distance from the agent and the objects. It can also be distorted by each environment and each class of the agent.

We tested several agents with the proposed algorithm to apply path integration. Initially, the agent has a random movement that depends on a random function based on uniform distribution. The movement of the agent can be designed as a monotonous



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

Figure 3.12: The results of the four different tests

course because the average of the uniform distribution is zero. A dynamic track of the agent is made by applying a two wheel system of the agent to the original model. First, the output of the two wheel system can be transformed to the accumulation based on the unit classes of the agent. Second, estimation of the accumulation can work to motor the agent. The results of this movement can be regarded as diverse cases of the track. Several examples of the tracks are shown in Figure 3.12.

Even though the three class method of the agent can be effective in reducing external noise, wrong optical flow vectors can cause inaccurate results in the estimation. In Figure 3.12, the track of the agent has some errors that arise from external noise. We checked the error of the each class for the three and five cases to investigate erroneous factors.

The result matrix is shown as Tables 3.4, 3.5 and 3.6. In the three class case, the estimation classes can usually be matched to the real classes. On the other hand, some problems arise in recognizing the existence of accurate classes in the five class case. The agent has difficulty distinguishing certain patterns that have the same motion but different sizes. Distortion of the panoramic camera can also occur and create an optical flow with the wrong direction. Although the agent can move translationally, each pixel cannot be matched translationally because of the curvature effect of the panoramic image transformed from the omni-directional image. Enhancement for performance can be achieved by reducing the transforming error of the omni-directional camera.

From these results, we can deduct precision, recall and F-beta values. These equations

	Real Results			
Category	1	2	3	Missing
Estimation	78	2	0	0
	0	51	0	0
	0	2	72	0
	0	0	0	8

Table 3.4: Result matrix for several accumulative tests: There are three main classes in summing methods such as left rotating with 5 degrees (category 1), straight forward movement (category 2), right rotating with 5 degrees (category 3). Missing category implies that two images taken from the snapshot model cannot make the optical flow vectors.

	Real Results			
Category	1	2	3	Missing
Estimation	71	10	0	0
	0	41	0	0
	0	11	61	0
	0	0	0	6

Table 3.5: Result matrix for several accumulative tests: There are three main classes of counting methods such as left rotating with 5 degrees (category 1), straight forward movement (category 2) and right rotating with 5 degrees (category 3). Missing category implies that two images taken from the snapshot model cannot make the optical flow vectors.

can be described as follows Sebastiani (2002):

$$P = \frac{\sum_{i=1}^{N} \frac{TP_i}{TP_i+FP_i}}{N} \tag{3.5}$$

$$R = \frac{\sum_{i=1}^{N} \frac{TP_i}{TP_i+FN_i}}{N} \tag{3.6}$$

$$F_\beta = \frac{(\beta^2+1)(P*R)}{\beta^2 P+R} \tag{3.7}$$

where P, R and F_β are precision, recall and F-beta values, respectively. N is the number of the classification cases. We use $\beta = 1$ cases of the F-value. The $\beta = 1$ means that the precision and recall for each case are regarded as the same weight for the impor-

	Estimation					
Category	1	2	3	4	5	Missing
Real	34	10	0	0	2	0
	0	16	0	0	0	0
	1	0	81	0	0	0
	0	0	0	11	0	0
	0	0	0	13	41	0
	0	0	0	0	0	5

Table 3.6: Result matrix for several accumulative tests: There are five main classes of summing methods such as left rotating with 10 degrees (category 1), left rotating with 5 degrees (category 2), straight forward movement (category 3), right rotating with 5 degrees (category 4) and right rotating with 10 degrees (category 5). Missing category implies that two images taken from the snapshot model cannot make the optical flow vectors.

	Precision	Recall	F-beta
3-sum	0.987	0.982	0.984
3-count	0.931	0.915	0.923
5-sum	0.914	0.833	0.872

Table 3.7: The results of precision, recall and F-beta values for each cases such as 3-sum, 3-count, and 5-sum methods, respectively.

tance of the performance evaluation. The table of results for each case is introduced as Table 3.7.

In Table 3.7, three-sum and five-sum methods, respectively, set the threshold value to distinguish three and five patterns by using summation of optical flow. The three-count method sets the threshold value to distinguish three patterns by using the number of counted optical flows with the same direction.

The results of angle and distacne errors for two cases such as three classes and five classes are shown in Figure 3.13. The amount of error is moderate when applied to real environment in the three class case. In the five class case, the agent has a slight difficulty in understanding 5 degrees and 10 degrees, so the threshold policy of the five class case must be further researched to improve the performance of path integration.

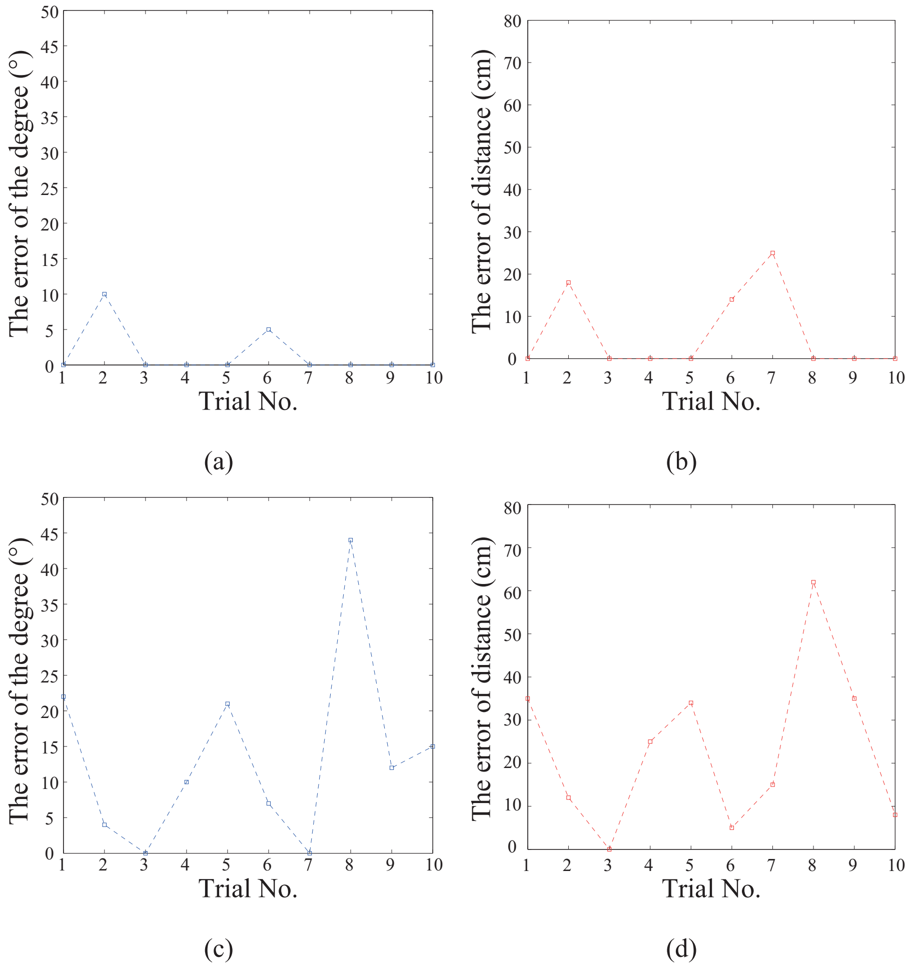


Figure 3.13: Several results of the angle and distance error for two cases which has 3 classes (a), (b) or 5 classes, (c), (d), respectively.

3.5 Summary of Chapter 3

Accumulative navigation is a basic model of local visual navigation using a snapshot assumption. For effective accumulated visual patterns, the agent has a summing operation with flow vectors on an omni-directional image made from the KLT algorithm. This operation results in a pattern of FOE & FOC that can estimate the moving direction of the agent from its reference position. The range that can detect correct patterns for agent movement can be extended by transforming a panoramic image from an omni-directional image. Direct use of the original omni-directional image can help to reduce the processing time needed to operate the optical flow algorithm, but this can create errors in the optical flow algorithm because of distortion of the lens of the omni-directional camera. In chapter 4, we will introduce methods that use panoramic images to find the homing direction.

Chapter 4

An ALV method based on the KLT tracker

In local visual homing navigation, the homing direction can be computed by measuring the disparity of intensity between two images taken at different locations. Several methods are available to estimate the disparity of the two images derived from the snapshot hypothesis, which uses only natural images of different locations to lead the agent to the home from its current location. Differential flow methods are a good choice for estimating the directional approaches using optical flow algorithms such as first or second order. In general, however, this method is known to be inferior to matching methods that do not feature preselection using a block matching algorithm for finding direction. Here, we investigate the combination of possibilities of a differential flow method using the Kanade-Lucas-Tomasi (KLT) tracker and a sector approach to improve the other performances, such as robustness and generality. In this paper, we suggest an improved model by controlling the number of corner points, the manner of setting the sector and the estimation of the orientation with the optical flow algorithm. This method has several advantages that save time, maintain good performance, and operate well without a compass in the general environment. This section is to be prepared for submission as a scientific paper (Cha and Kim, 2013d).

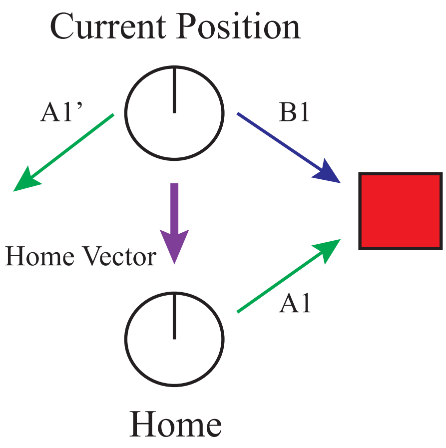


Figure 4.1: Average Landmark Vector (ALV) method

4.1 Combination of ALV method and KLT tracker

The KLT tracker is applied to the ALV algorithm using a start point and end point in each vector. The start point refers to the angular value of each vector at a reference position while the end point refers to the angular value of each vector at the current position. In the ALV method, the size of the vector in each image is not useful, but the angular value of each vector is important for estimating the homing direction. From these results, all vectors with a size regarded as a unit from the equal distance assumption (Franz et al., 1998) are summed at each position to make a representative vector. From the equal distance assumption, the size of each vector can be regarded as a unit, so the homing vector is calculated as:

$$\overline{h} = \overline{ALV_{tar}} - \overline{ALV_{cur}} \tag{4.1}$$

where $\overline{h}$ means the homing vector. Figure 4.1 shows the ALV method to estiamte the homing vector.

The $\overline{h}$ and $\Delta\mathbf{v}$ of results from the KLT tracker can be defined by the ALV. The result of the KLT tracker is a vector made from the start point of each corner. According to the ALV, the deviation of landmarks between two images can be regarded as a descriptor of the representative vector. Assuming $\mathbf{p}$ is a unit vector that indicates the position of a corner detection algorithm, $\mathbf{p_i}$ and $\mathbf{p_i} + \Delta\mathbf{v_i}$ are representative vectors for each position. In contrast, $\mathbf{p_i} + \Delta\mathbf{v_i}$ is not an unit vector generally, so $\mathbf{p_i} + \Delta\mathbf{v_i}$ must be normalized before calculating the homing vector $\overline{h}$. On the basis, $\overline{h}$ can be derived as the following:

$$\overline{h} = \sum_{i=1}^{N} \frac{\mathbf{p_i} + \Delta\mathbf{v_i}}{\| \mathbf{p_i} + \Delta\mathbf{v_i} \|} - \sum_{i=1}^{N} \mathbf{p_i} \tag{4.2}$$

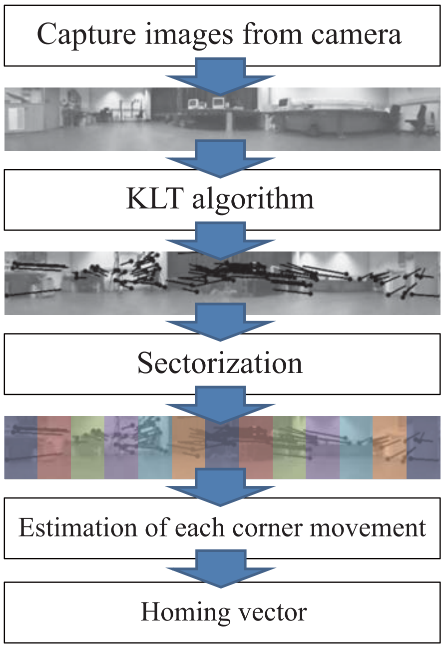


Figure 4.2: Introduction of proposed model which estimates the movement of the agent

where $\overline{h}$ is an estimate of the homing route derived from a combination of the ALV method and the KLT tracker.

Figure 4.2 indicates the whole algorithm of the proposed model to estimate the movement of the agent. First, we capture the two images from two locations: reference and current. Next, the KLT algorithm calculates an estimation of the movement of each vector from several corners between the two images. We can observe the total movement of the vectors from the sum of vectors multiplied by weights at each sector. From this information, we can calculate the homing direction from the current location to the home location.

4.2 Orientation of the agent based on the KLT tracker

The orientation issue is also a very importance problem for differential flow methods because these approaches are not guaranteed to align angles automatically from the result vectors of optical flow. We introduce an orientation method to minimize the error of the aligned angle. Equation 2.9 provides an easy calculation by assuming a rotational factor R to zero. Orientation means that this assumption can be retained after removing the rotational movement of the agent and camera with respect to corner features. We suggest a new approach using the LK algorithm to minimize the rotational component

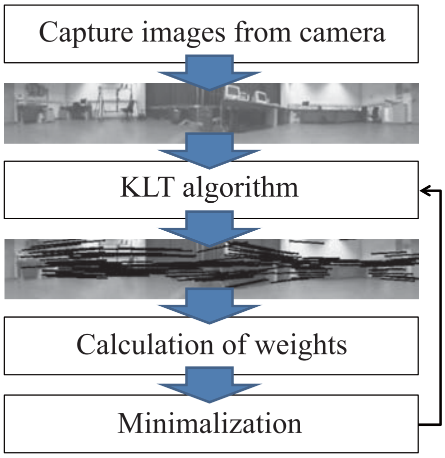


Figure 4.3: Introduction of proposed model which aligns angle of agent

of optical flow in this paper. First of all, the agent located in a certain position without orientation can capture the image of the current state. The LK algorithm then makes a set of optical flow vectors with two images, such as the current and home locations. The image taken at the current location shifts the unit angle pixel left or right and repeats the process, thereby creating optical flow. If the minimum shifting position is found, the shifting value represents the distortion of the angle of the agent from the reference angle of the home location, so we can make a distorted equation E as follows:

$$E(\theta) = \arg\min[\mathbf{v}(\theta)] \tag{4.3}$$

where θ can be ranged from 0°~360°with same interval. Compared with the previous results (Zeil et al., 2003; Labrosse, 2006; Burke and Vardy, 2006), this algorithm gives more accurate results for some cases than does any of the other methods.

A precise description of the orientation method based on KLT algorithm is well represented by Figure 4.3, which shows the whole algorithm. Basically, it is similar to finding the homing direction based on the KLT algorithm. The characteristics of the orientation algorithm include an additional process that checks the sum of weights. For an accurate calculation, a large size vector is regarded as having a large error of alignment angle and corners that cannot be drawn. This vector has high weight because the KLT algorithm results in a vector within the condition where the corner has a proper error in a range made initially by the user. This process is iterative for all angles until the minimum error angle is found. This angle can be chosen as the angular position of alignment of the agent.

(a)

(b)

Figure 4.4: Example of ALV sectorization: (a) is divided into four sectors. (b) is divided into eight sectors.

4.3 ALV sectorization

The combination of the sector approach and KLT algorithm can be a good alternative choice because it has more advantage of the effectiveness than any other algorithms by controlling several parameters. Figure 4.4 shows an example of the ALV sectorization. The number of divisions of sector means a resolution of images for angle. We find that the performance of proposed algorithm is robustness for the small number of corners and the low resoultion of sectors. This condition can help to reduce the computation time of algorithm to estimate the homing direction of the agent.

The sectorization is inspired from change of the each photoreceptor in compound eye of the insect. In sector method, the image segmetation for angle is essential to know current state of the landmarks. nitially, each image is divided as several sectors to measure alternation for distribution of corners. In each image, this algorithm stores the number of corners per each sector for two images, respectively. The difference of the number of corners in sector means a movement of the agent relatively.

4.4 Comparison with other methods

The representative methods that can substitute for the KLT tracker are the sum square difference (SSD) algorithm and the block matching algorithm. The block matching algorithm was used by Vardy and Möller (Vardy and Möller, 2005) to estimate the homing direction of an agent. They were able to estimate the distance from objects through a block matching method and used a combination of the differences between a reference image and the current image, distance from objects, and the FOE and FOC

(a)

(b)

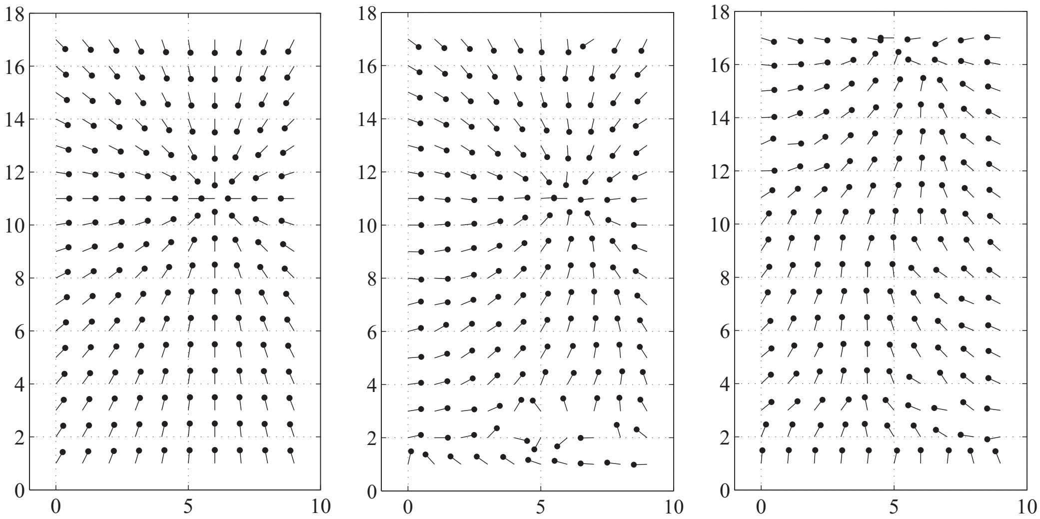
(c)

Figure 4.5: Test results for other methods: (a) is original method. (b) and (c) are result of block matching algorithm and templete matching algorithm, respectively.

patterns. The block matching method is robust for uncertainty noise and illumination changes in the image but this algorithm can be slow exponentially when the window size block matching algorithm is increased. Precise results depend on having a large window size for the ALV algorithm due to the lack of visual information made from the snapshot assumption.

The SSD algorithm seems to be an alternative way to overcome the limitations of the block matching method. Template matching of a certain area around the corners is used to match other areas in an image as an efficient method to find similar areas, but this method has a drawback in that it cannot consider the FOE and FOC distortion made by an omni-directional camera. Therefore, the determination of the homing direction can be drastically degraded when the agent finds a homing direction at a certain position far from the reference position.

Figure 4.5 indicates results of other methods used to estimate homing direction. Both cases can successfully create optical flow vectors between two images, but the KLT has critical advantages for moderate performance and fast speed compared with these other algorithms. Although the accuracy of the homing vector in block matching is better than first order or second order methods (Vardy and Möller, 2005), the speed of processing in the KLT tracker is faster than block matching. For example, the average processing time for the KLT tracker is 0.02s whereas the average processing time for block matching is 0.5s in some tests. This gap can have critical effects on synchronization of the measurement model and the control model in real cases.



(a) AAE: 0.0000, RR: 1.0000 (b) AAE: 0.3201, RR: 1.0000 (c) AAE: 0.3963, RR: 1.0000

Figure 4.6: Ideal case (a), proposed model case (b) at home position (7, 11) and case (c) at home position (5, 17) with all conditions that the number of corners is 500 and the number of sectors is 200.

4.5 Experiments and results

4.5.1 Experimental procedures

Our experiment to find a homing direction used the databases of images by Vardy and Möller (Vardy and Möller, 2005) that are existed online. that are available online. Images of the experiment are taken from an omni-directional camera of the agent at a certain position and are converted from omni-directional images to panoramic images. Finding a vector on the panoramic image helps to extend the range of observable directions correctly compared to the omni image. We tested two image sets: the original set and the hall1 set. Images of the original set consisted of dimensions of 5.5 $\times$ 8.25m on the computer lab. The real capture area on the original set was 2.7 $\times$ 4.8m on the grid, with 30cm resolution. Images of the hall1 set contained a capture area of 4.5 $\times$ 10m at 50cm resolution.

Initially, all images of each set were oriented in one direction for description using a compass. However, to build an experiment without a compass, each image was rotated by a random angle, so a combination model of finding a direction and orientation used different conditions compared with experiments with a compass.

Basically, coding of the proposed model is designed by a parallel model. The Pthread library in Linux can help to divide parts of the proposed model as control and measurement models. Initially, both models are independent of each other but, to prevent mixing of controls and measurements when the agent takes a picture at a certain position, a semaphore algorithm is activated to control the working priority. That is, during a measurement state where a picture can be taken, the control state is put to sleep until the measurement state is ended. A parallel process can help to upgrade performance through hardware control because the series model must shut down other parts to control certain parts and ensure that the speed of the control is faster than the series model.

4.5.2 Experiment with a compass

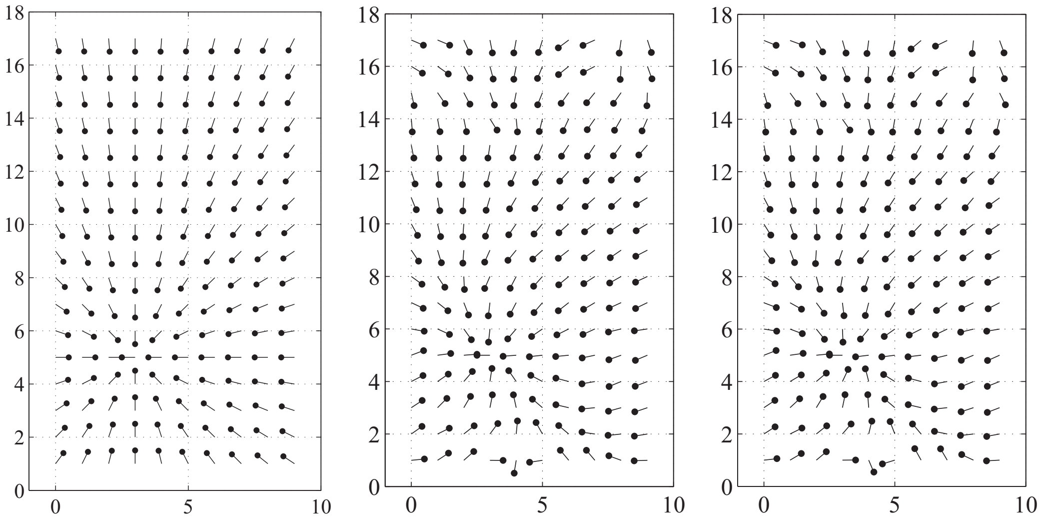
4.5.2.1 Fixed value case

The homing vector of the fixed value case with a compass can be made from the following:

$$h(x) = -\sum w_i \vec{b}_i - \sum w_j \vec{b}_j \tag{4.4}$$

where x is the real position of the agent on the map. We test this model for homing performance of the agent in two environments. We use two criteria: the average angular error (AAE) and the return ratio (RR), as used previously Vardy and Möller (Vardy and Möller, 2005) to measure the performance of the homing method. The AAE is the average angular distance between the measured home vector $\hat{h}$ nd the ideal home vector h at the current position. The RR can be computed by checking the possible places for homing when the agent moves along the homing vector at each position. If the agent makes a successful return to the home position, the RR is increased. The RR can be calculated as the ratio of the number of successful homing places to the total number of all places on the map.

Figure 4.6 is an example of a fixed value model and compares this model and the ideal case at the home position (7, 11). Even if a small part of the homing direction map is distorted, zero of the RR in the proposed model case means good performance for homing. If the home position is located at the edge of the map, the performance of homing is inferior to cases of the general home position, but the RR is still 1 compared with the (5, 17) case.



(a) AAE: 0.0000, RR: 1.0000 (b) AAE: 0.2861, RR: 0.9722 (c) AAE: 0.2908, RR: 0.9667

Figure 4.7: Ideal case (a), ALV case based on KLT algorithm (b) and proposed model case (c) at home position (4, 5) with conditions that the number of corners is 500 and the number of sectors is 200.

We confirm the performance of the proposed model by comparing the ALV and proposed model as shown in Figure 4.7. The AAE and RR are similar in both cases, but ALV based on KLT takes more time than the proposed model because ALV uses all landmarks as corners on the image, compared with the sector case that uses only certain landmarks selected from the feature detection algorithm. The proposed model maintains robustness of performance when the number of corners and sectors is small. The experiments shown in Figure 4.6 and Fig. 4.7 used the original environment set.

Figure 4.8 shows good results for all cases that modulate two main parameters such as the number of sectors and corners. Color and brightness indicate the AAE of the homing performance with each condition. The distribution of color indicates that this model can reduce time by removing computation of the LK algorithm for many feature points, as $O(nN+n^2)$ (Baker and Matthews, 2004).

The examples of cases for diverse number of parameters are given in Figure 4.9. From these results, we see that the number of sectors is more sensitive than the number of corners in maintaining the performance of the homing for agent in these cases. That is, the resolution can be regarded as an important factor in estimating the homing direction.

Figure 4.10shows two graphs that represent AAE with respect to two main parame-

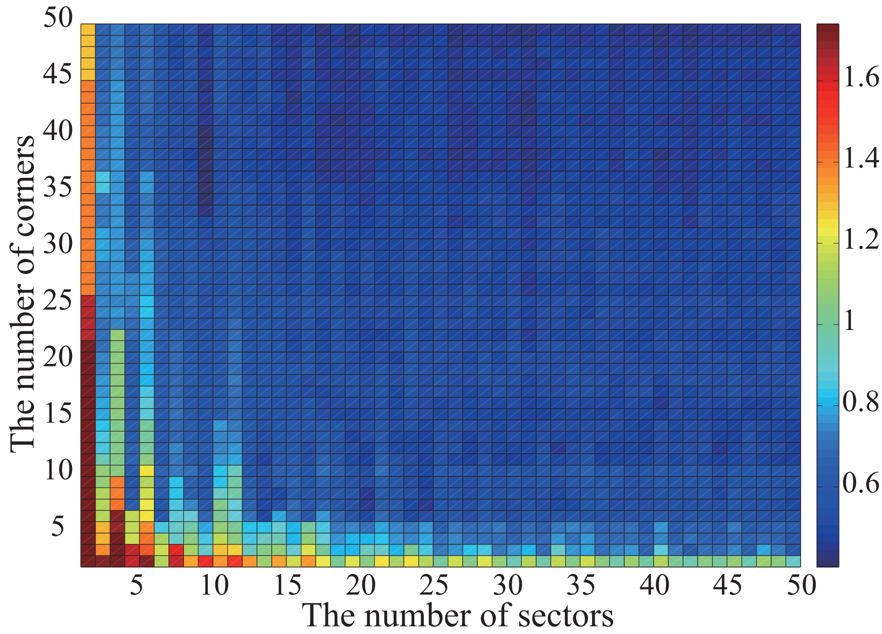
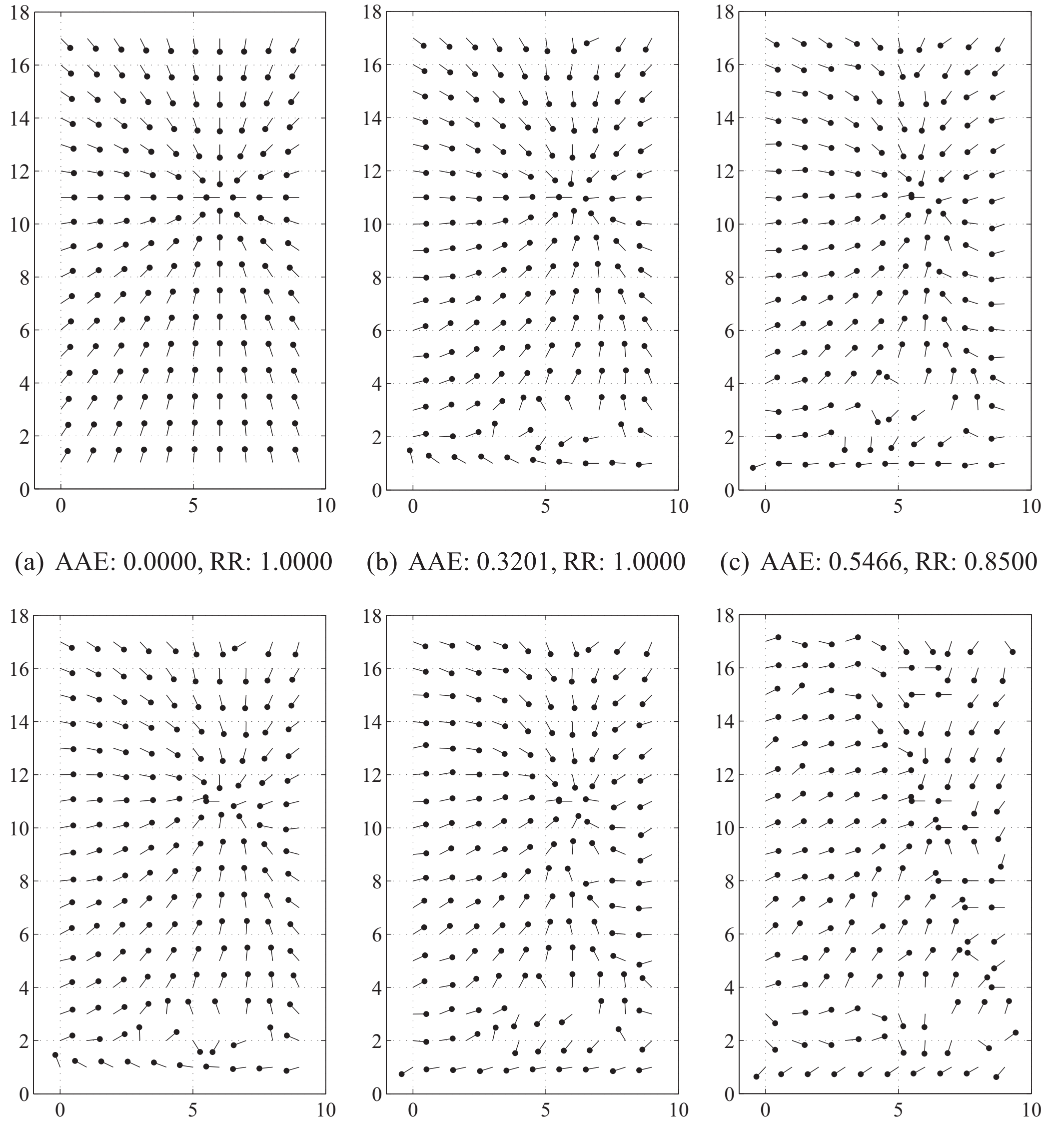


Figure 4.8: The AAE of homing with respect to two parameters such as the number of sectors (x axis) and corners (y axis).

ters: the number of corners and sectors. The AAE declines rapidly as the number of parameters increases. This means that the model based on the KLT and sector does not need too many components for calculation of the movement to return home. This result seems to contradict previous results in Figure 4.9. But, as shown in Figure4.8, for a very small range of sectors, the number of corners can also be particularly sensitive for performance of the homing process.

The feature detection algorithm helps to maintain the stability of the homing performance. Before checking, we select three cases for selecting corners that use a dynamic method, a fixed method, and a random method. The dynamic method is one that we suggest as a good feature detection algorithm. The fixed method is one where corner positions are fixed regularly, the same as with the block matching method. The random method is one where the corner positions are randomly selected. We find that introduction of a selecting method into the proposed algorithm serves to reduce the deviation of the distribution of performance at all positions in the two environment cases: original and hall. The results of distribution for AAE for methods and environments are shown in Figure 4.11 and Figure 4.12.



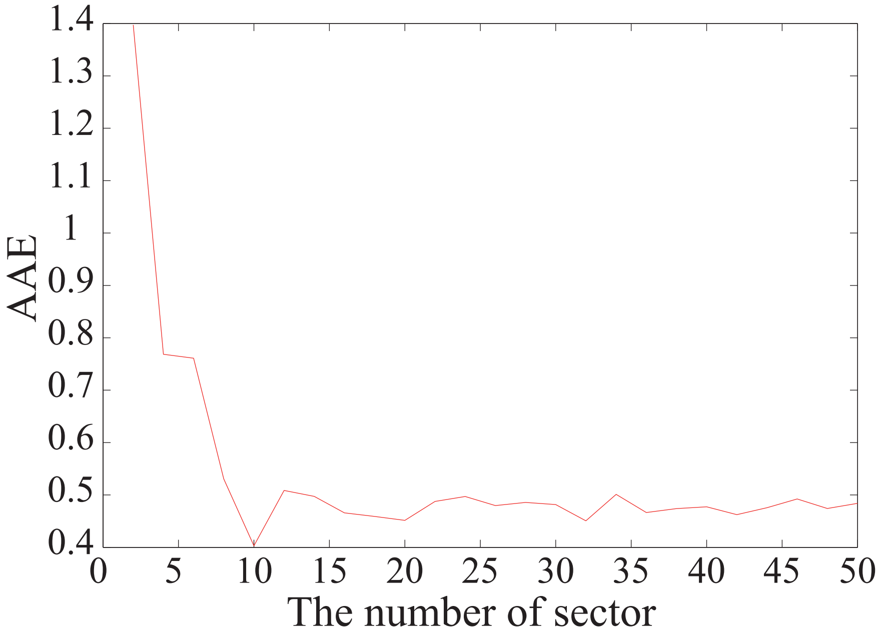
(a) AAE: 0.0000, RR: 1.0000 (b) AAE: 0.3201, RR: 1.0000 (c) AAE: 0.5466, RR: 0.8500

(d) AAE: 0.3420, RR: 1.0000 (e) AAE: 0.5545, RR: 0.8556 (f) AAE: 0.8073, RR: 0.7706

Figure 4.9: All the proposed models are given as (a)~(f) with different conditions. Ideal case (a), the number of corners is 500 and the number of sectors 200 (b), case where the number of corners is 200 and the number of sectors is 40 (c), case where the number of corners is 40 and the number of sectors is 200 (d), case where both of the number of corners and sectors is 40 (e), and case where both of the number of corners and sectors is 20 (f), respectively in original environment.

4.5.2.2 Adaptive value case

Even though a fixed value case can be regarded as establishing the model to reduce the number of corners and sectors, the effectiveness of reducing sectors is meaningful

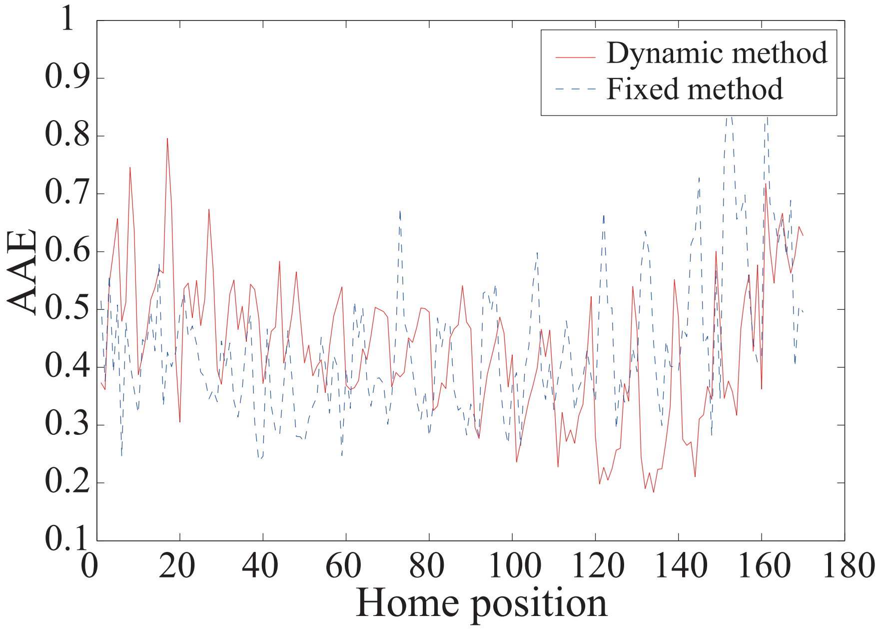


(a) AAE versus corner(The number of sector is fixed as 10.)

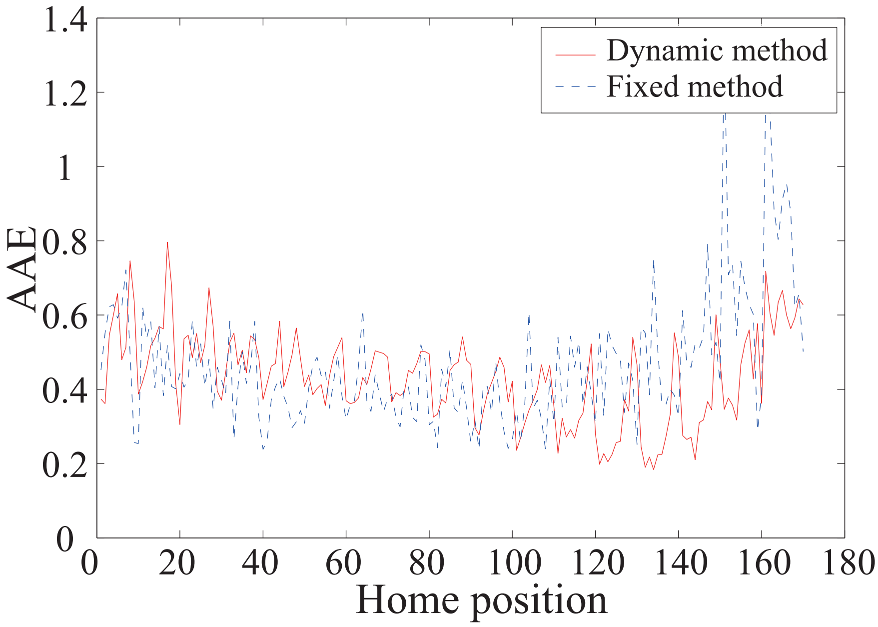
(b) AAE versus sector(The number of corner is fixed as 35.)

Figure 4.10: Two graphs show that proposed model has characteristic of fast convergence for AAE with respect to the number of corners (a) and sectors (b) in original environment.

when the number of sectors can be dramatically reduced. The adaptive value case is a good choice with respect to the number of sectors. Unlike the fixed value case, the



(a) AAE for dynamic method (Red line) and fixed method (Blue dot)

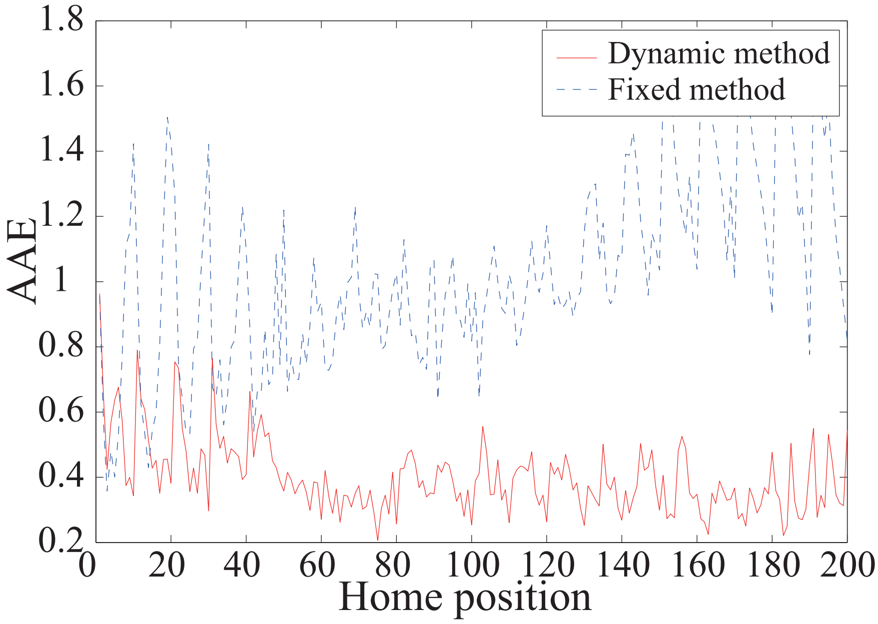


(b) AAE for dynamic method (Red line) and random method (Blue dot)

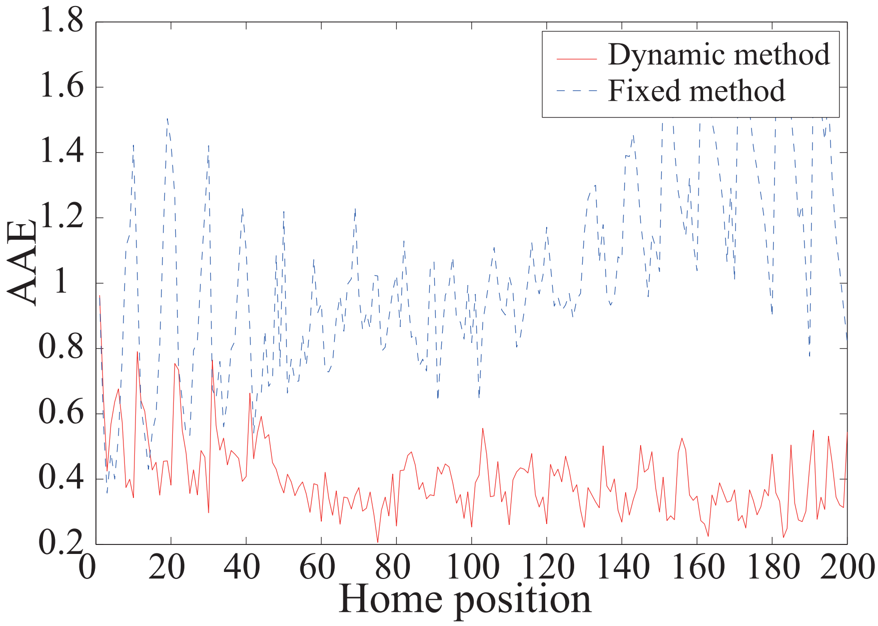
Figure 4.11: Comparing of AAE for all original environment positions using three methods which makes setting corners in original environment: dynamic versus fixed (a), dynamic versus random (b).

adaptive value case requires a somewhat more complex equation, as follows:

$$h(x) = -\sum w_i \vec{b}_i - \sum w_j \vec{b}_j \tag{4.5}$$



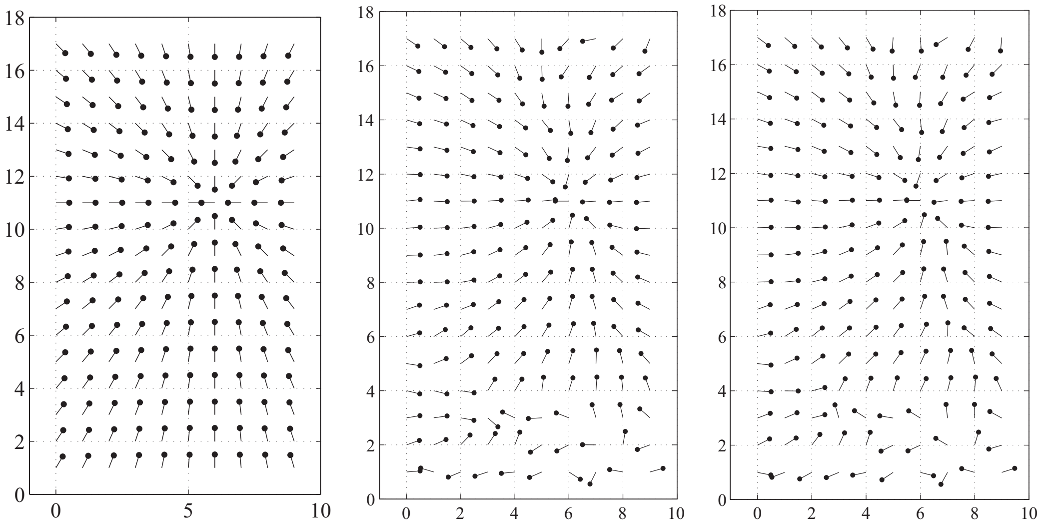
(a) AAE for dynamic method (Red line) and fixed method (Blue dot)



(b) AAE for dynamic method (Red line) and random method (Blue dot)

Figure 4.12: Comparing of AAE for all original environment positions using three methods which makes setting corners in hall environment: dynamic versus fixed (a), dynamic versus random (b).

where $N_{i,left}$ and $N_{i,right}$ can be checked by confirming the distribution of corners in each sector. This algorithm focuses on the center of mass for all corners rather than the average value in each sector, so when the weight of each factor is calculated, the



(a) AAE: 0.0000, RR: 1.0000 (b) AAE: 0.4442, RR: 0.9222 (c) AAE: 0.4206, RR: 0.9444

Figure 4.13: ideal case (a), adaptive value cases at home position (6, 11) with conditions when the number of corners is 50 (b) and 100 (c). The number of sectors are 4.

weight can be affected considerably by the performance of corner detection.

Figure 4.13 shows that the adaptive value case has good performance for a case with 4 sectors. That is, increasing the AAE from a reducing sector is not a mechanism for sectorizing the image but setting the problem of the representative value on each sector. Even if the number of corners and sectors is small, the AAE and RR of the homing performance are moderate in the adaptive value case.

4.5.3 Experiment without a compass

The orientation problem of correspondence methods is very important to accommodate the uncertainty of controlling the agent. Generally, an external compass such as a magnet helps to align the agent, but if the agent encounters an environment that cannot use an external compass, a visual compass has to be employed to maintain homing performance. The linear search used by Labrosse, which included phase correlation and an improved version to save processing time where the linear search was added as a sample search (Labrosse, 2006; Burke and Vardy, 2006), is a good example of a visual compass. In this paper, we introduce a new alignment method using the optical flow algorithm and flow search. The basic concept of this new method is similar to linear searching for the evaluation of similarity between two images. If the agent is

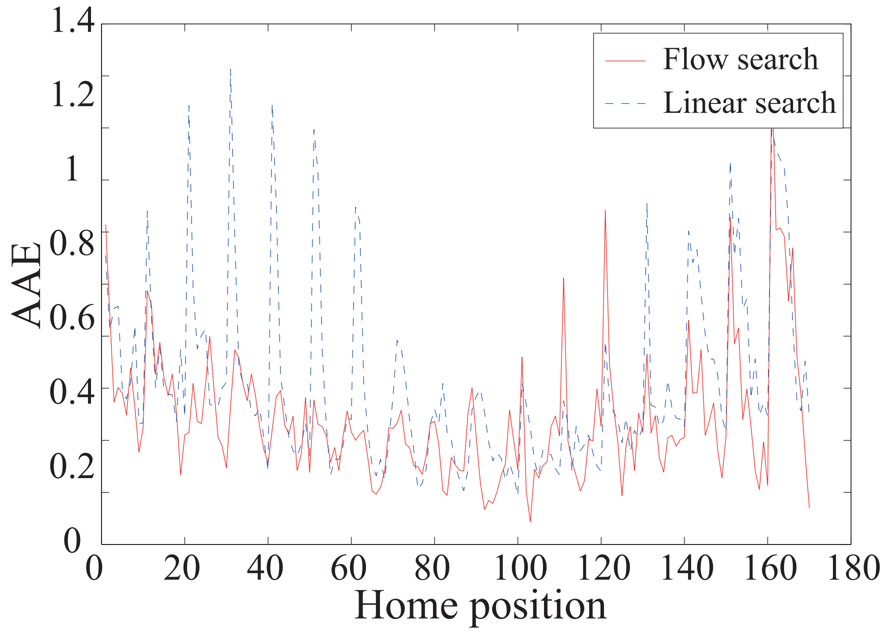
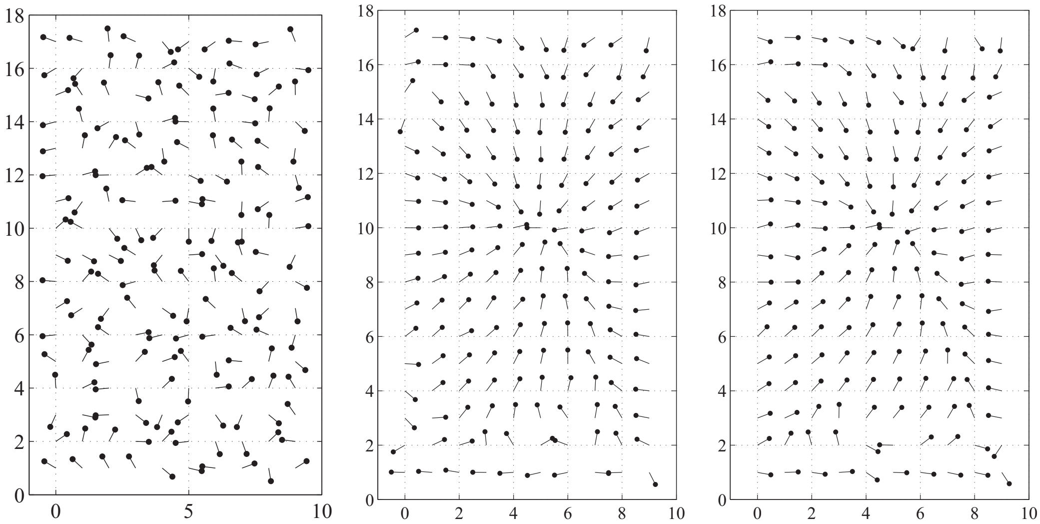


Figure 4.14: Comparing AAE for both cases which are flow search(Red line) and linear search(Blue dot) in whole range.

aligned, the rotational component from equation 2.9 is almost zero. This means that the result of the optical flow vector after calculating the LK algorithm on the image is a minimum value when the agent looks in the correct direction.

The novelty of this method is the improvement of accuracy for finding the alignment angle with an estimation process based on the optical flow algorithm rather than a calculation of the difference for all pixels. Even if more time is spent than with a linear search because of heavier relative calculations for the optical flow algorithm, a sample search (Burke and Vardy, 2006) can reduce the time difference between the flow search and linear search.

We compare our method and the linear search method. Because the phase correlation method has similar performance to that of a previous linear search (Burke and Vardy, 2006), we can omit the testing phase correlation to compare performance. Figure 4.14 shows a comparison of the performance for a combination of finding direction and orientation with the flow search and linear search in the original environment. The flow search is better than the linear search with respect to homing performance. Previous research (Burke and Vardy, 2006), used only 9 by 9 cells for a 17 by 10 original environment. We guess that limitations of location range for good performance exists at approximately 9 by 9 cells from the homing location. The flow search has also a limita-



(a) AAE: 1.4537, RR: 0.0000 (b) AAE: 0.4265, RR: 0.9388 (c) AAE: 0.4387, RR: 0.9222

Figure 4.15: Non-orientation case (a) and homing direction maps at home position (5, 10) when agent do not have an external compass: flow search (b) and linear search (c).

tion in the location range similar to that of the other visual orientation methods. Therefore, for a more precise comparison, we confine that range of orientation to around 9 by 9 from the homing location. The performance result at this range is shown in Figure 4.14. Our method is also better than the linear search, as shown in Figure 4.14.

Real examples of homing direction maps constructed without an external compass are given in Figure 4.15. A pure approach based on the sector approach without a compass gives the worst results of all of the cases, so a compass must be needed. Both cases show moderate performance even though the agent with the camera has no external compass.

4.6 Summary of Chapter 4

The model proposed in chapter 4 has two significant features for local visual homing navigation. First, the proposed model can automatically find landmarks on a complex image to use in finding the homing direction. Second, finding a direction with this model consists of two concepts: the ALV method and the FOE and FOC method. Even though the equal distance assumption presents a critical problem for real cases, the performance at finding direction can be maintained in consistent conditions.

The idea of a sector is inspired by the visual cells of insects that have compound eyes. Despite the reduction of the number of neurons in an insect's brain, the insect can make a judgment with respect to finding a homing direction. Based on this idea, if the insect can use optical flow from its visual sensory system, the performance of algorithms made from the optical flow algorithm can be regarded as robust image resolution. Through several tests, we showed that the proposed model can operate well in homing navigation using low resolution images.

The model proposed in this chapter introduces a new problem; namely, that this model cannot detect external noise in the surroundings of the agent. Even though almost all of the test cases find the correct homing direction, wrong data can occur and give rise to a critical problem that can degrade the performance of this algorithm. To avoid distortion of the measurement model of the agent, we next introduce a probabilistic approach to visual navigation using parametric and non-parametric filters.

Chapter 5

Local visual navigation with a Bayesian filter

Agent localization is fundamental task in many cases of robotic navigation. In probabilistic robotics, high accuracy and a robustness algorithm based on a Bayes filter help to enhance the performance of agent localization. On the other hand, high complexity and pre-searching for localization remain as inevitable tasks. A bio-inspired model for homing navigation can be a good choice to solve these problems. Nevertheless, research into bio-robotics for localization problems has shown little progress when compared with engineering research. In this paper, we focus on a mixed model of probabilistic and biorobotic approaches for navigation in order to take several advantages from each field. The measurement model uses optical flow and is inspired by the vision system of the compound eyes of insects as a core idea for a biological approach to reduce complexity. From this measurement model, the dimensionality of vision sensing information decreases to one, which saves processing time. The agent location can be determined by applying a Bayes filter based on active sensing information using only two images taken from an omni-directional camera at the reference and current positions, even though the agent has no pre-searching information for unvisited positions on the map. This part is to be prepared as a paper for submission to a scientific journal.

This bio-inspired application of a local visual homing navigation is inspired by insects. Some insects use an optical flow system to find their homing direction within their environment. Even though they do not have a complex localization system, such as the place cells and grid cells common in mammalian neurons, they can find their routes with fairly reliable accuracy. They have adapted with systems with low capability for information storage and processing by developing distinctive mechanisms. For instance, their ability for local visual homing navigation using only intensity information can be regarded as one of the methods that can be used as a bio-inspired approach. Reduction of the amount of information helps to increase the processing speed needed to determine the homing direction at a certain position. Nevertheless, previous bio-inspired research has avoided taking a probabilistic approach to prevent disrupting efficient processing for returning home, even though this could help to increase the robustness of the homing performance in a noisy environment. In this section, we propose a new homing navigation method that takes a Bayesian approach to maintain efficient capacity of computation. To minimize the computational load of navigation, the Bayesian model is made of a one-dimensional Kalman filter based on Gaussian distribution. We confirm the homing performance of an agent with the proposed model in a real environment and demonstrate that the proposed model is robust in a noisy environment. This content has been prepared to a journal (Cha and Kim, 2013b) and submitted to a conference (Cha and Kim, 2013c).

5.1 Localization method with desired map

5.1.1 Descriptors for the measurement model

We made descriptors using the KLT algorithm based on the ALV method (Lambrinos et al., 2000). Since the ALV algorithm is needed for compass information, the distance-estimated landmark vector (DELV) model (Yu and Kim, 2011a,b) or the warping method (Franz et al., 1998; Möller, 2009; Möller et al., 2010) can also be considered for the measurement model. However, this case has a drawback in that it spends higher computational time than other methods. Therefore, we suggest the KLT algorithm based on the ALV method using information from a visual or an external compass. The ALV algorithm can estimate the homing vector by calculating the angular position difference of several landmarks. However, the corners made from the KLT algorithm are not constant over the whole map. So, unlike the meaning of the ALV algorithm, the difference in each vector reflects the distribution of the FOC and FOE between the two images (Vardy and Möller, 2005) as well as difference between each landmark. Nevertheless, results of this algorithm are moderate when applied to homing navigation because FOC and FOE also mean relative movement of the agent (Vardy and Oppacher, 2003, 2004).

Figure 2.8 describes the distribution of the FOC and FOE model on a panoramic image unfolded from an omni-directional image. These features can occur by distorting optical flow from the curvature of the omni-directional camera. Basically, optical flow is dependent on the velocity of the -agent and the distance from the agent and the objects observed, so the movement vector affects optical flow as well as the relative distance of the position for objects. For example, if the agent moves toward a specific direction, the camera range that is perpendicular for movement makes relatively longer flows compared with movement of agent at any other ranges. Overlooking of one of the two can result in erroneous conclusions regarding the estimation due to wrong corner matching, which finds incorrect correspondence when the agent automatically draws flow fields between two images with the optical flow algorithm. Consequently, consideration of two factors – movement of landmarks and distortion from the curvature of the omni-directional camera – helps to illustrate the movement of the agent more precisely.

From equation 4.2, the homing vector $\overline{h}$ in the local visual homing navigation with intensity information is used as a different meaning descriptor in agent localization.

In the measurement model of probabilistic approaches, the descriptor $\overline{h}$ is not always a homing vector because the noise signal can be added. If the added noises are none or small, the existing model of visual homing navigation works well. In a complex environment as a real case, the model is not guaranteed to work efficiently to find the homing position, but it is reasonable that values near the ideal vector, which indicates homing direction, can be observed more frequently than other values. To apply this idea to a Bayes filter, the measurement model can be assumed as $\Delta\theta_t$ which stands for the angle deviation between a vector observed in practice and a desired vector, which can be watched at the current position without any noise. The measurement model $\Delta\theta_t$ can be expressed as the following:

$$\Delta\theta_t = |\,\overline{h}(\theta) - \widehat{h}(\theta)\,| \tag{5.1}$$

where $\widehat{h}$ is the ideal vector at each position. The measurement update can then be derived as the following:

$$Bel(x_t) \propto P(z_t|x_t)\overline{Bel(x_t)} = P(\Delta\theta_t|x_t)\overline{Bel(x_t)} \tag{5.2}$$

Change of measurement update as $\Delta\theta_t$ from z_t works on dimensional reduction for the measurement vector. It is particularly important that $\Delta\theta_t$ is only one-dimensional information. This can help to reduce computational load to activate discrete filtering.

In a discrete filter, each grid has each an uncertainty from the update model, so the measurement model can be rewritten as the following:

$$Bel(x_t^k) \propto P(\Delta\theta_t^k|x_t^k)\overline{Bel(x_t^k)} \tag{5.3}$$

where k is the label of grid. If the agent uses a particle filter instead of histogram filter, k can be regarded as the label of a particle.

In equation 5.3, the measurement update $P(\Delta\theta_t^k|x_t^k)$ can be designed as a Gaussian distribution with an average of zero. When $\Delta\theta_t^k$ is zero, the probabilistic output of $P(\Delta\theta_t^k|x_t^k)$ s mostly higher than in other cases. When a more correct visual sensor is used, the smaller value of the standard deviation can be applied to the Gaussian distribution. In a particle filter, $P(\Delta\theta_t^k|x_t^k)$ is used as an important weight for criteria to decide the sample frequency in the resampling phase. From this, the measurement model can be expressed by using Gaussian distribution as follows:

$$Bel(x_t^k) = \eta N(\Delta\theta_t^k|\mu,\sigma)\overline{Bel(x_t^k)} \tag{5.4}$$

where μ is the average and σ is the standard deviation of the Gaussian distribution. If the difference of angle $\Delta\theta_t^k$ does not have a specific bias or offset, the average is fixed as zero. Standard deviation σ can be selected as a suitable value for an adapting environment. $N(\Delta\theta_t^k|\mu,\sigma)$ can be represented by the Gaussian equation as the following:

$$N(\Delta\theta_t^k|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(\Delta\theta_t^k-\mu_t)^2}{2\sigma^2}} \tag{5.5}$$

Figure 5.1 shows an example of a Gaussian distribution for the measurement model where the average is zero. If a sensor can be regarded as accurate, the standard deviation σ of the Gaussian distribution is small. This means that the higher the reliability of the measurement update, the more the weight of the measurement update $P(\Delta\theta_t^k|x_t^k)$ influences the belief of the agent state $Bel(x_t^k)$.

5.1.2 Active sensing strategy for a control model

Active sensing strategy originated from biological approaches. In contrast with passive sensing, active sensing is where the agent can get several signals while generating self

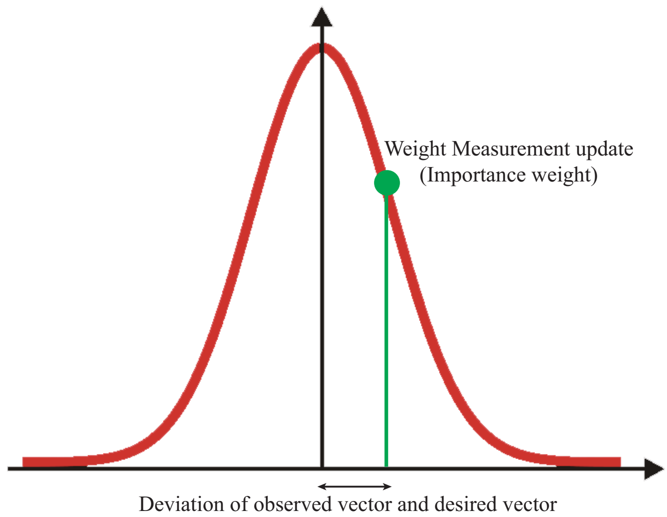


Figure 5.1: Example of Gaussian distribution for measurement model: Intuitively, updating weight can be represented with Gaussian distribution for deviation of observed vector and desired vector. Generally, average of Gaussian distribution used for measurement update is zero. Standard deviation of Gaussian distribution can be changed according to the reliability of sensory system.

energy. In this paper, we will show that localization performance of an agent with active sensing is better than the general case with passive sensing. To establish a system of active sensing, the agent must be needed for the reference angle for orientation in order to recognize its own direction of movement that comes from its self-energy.

Without an active sensing strategy, an orientation task becomes a crucial problem in local visual homing navigation. In particular, the ALV and differential flow approaches require an external compass. Even if an external compass is provided, some cases that cannot use GPS sensors or magnetic compasses can exist, so visual compass methods can be alternative solutions in specific environments. Generally, the incorrect working of an external compass occurs occasionally on maps with mobile robots in indoor environments.

Labrosse suggests a linear search method to find the reference angle (Labrosse, 2004). This method is a representative approach using visual compass methods. The linear search method is based on the assumption that pixels of an image vary smoothly and monotonically with spatial distance (Zeil et al., 2003). This method is simple. Pixels of two images at the same position are subtracted and the differences are accumulated until all the pixels are subtracted to get total Euclidian distance for all angles. An agent can find a distorted angle from a reference angle at a certain angle with a minimum accumulated error.

To improve processing speed, Burke and Vardy proposed two methods; namely, phase correlation and the sample search approach (Burke and Vardy, 2006). Phase correlation was created by Kuglin and Hines (Kuglin and Hines, 1975) and uses the axis of a fast Fourier transform (FFT) and compares the similarity of patterns between two images. Even if FFT is a complex computation, since a digital signal processing (DSP) processor helps to accelerate processing speed, phase correlation is faster than linear search in some cases.

Sample search is variation of the linear search where processing is improved by choosing a minimum angle that is faster than the original version. First of all, sampling from a whole angle distribution with regular intervals is done and the chosen corners are compared with each other. If the corner with a minimum error is found, sampling is repeated around that found corner as a center with regular intervals where size is half that of the previous intervals. This process can be iterated with a specific threshold. The phase correlation method and the sample search method cannot outperform the linear search in several tests but processing time can be reduced rapidly.

After orientation of the agent, an active sensing strategy increases in effectiveness at controlling the model for the probabilistic approach. In a desired grid-based map, the agent can choose eight directions when it encounters the control phase. Active sensing strategy with compass information is activated by deciding the direction of the agent. This is strong evidence to believe in a direction of movement for the agent. Therefore, we can design a control update $P(x_t|u_t, x_{t-1})$ with the probabilistic equation, as follows:

$$P(x_t|u_t, x_{t-1}) \propto e^{-\frac{(u_t-d_t)^2}{k_w}} \tag{5.6}$$

where d_t is the observed direction of movement with compass information at time t, k_w is a width constant for place and direction. This function is affiliated with a Gaussian function. It is meaningless that the normalizing process in the control update because the normalizer η is already located in the measurement update part. To adapt a discrete filter, the control update can be designed by a grid parameter k as follows:

$$\overline{Bel(x_t^k)} = \sum_{k=1}^{K} P(x_t^k|u_t^k, x_{t-1}^k) Bel(x_{t-1}^k) \tag{5.7}$$

where K is the total grid or particle number. Therefore, this equation can be rewritten

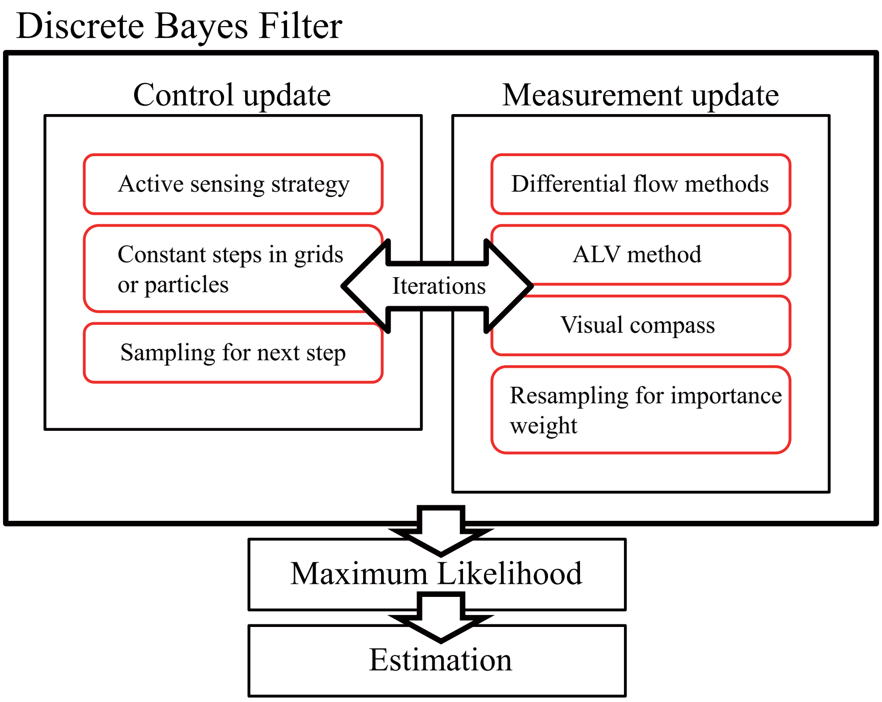


Figure 5.2: Whole system structure of proposed algorithm: For some iterations, maximum likelihood can be converged to suitable values in discrete Bayes filter. Control and measurement update can increase the accuracy of proposed algorithm. After finding the maximum likelihood, agent can estimate its own position using maximum likelihood.

with a Gaussian model as the following:

$$\overline{Bel(x_t^k)} \propto \sum_{k=1}^{K} e^{-\frac{(u_t^k - d_t)^2}{k_w}} Bel(x_{t-1}^k) \tag{5.8}$$

Figure 5.2 shows the system structure of the algorithm proposed in this paper. In the Bayes filter, parts of the measurement update and control update can be designed by local visual homing methods with intensity information as differential flow methods and active sensing strategy, respectively. Through several iterations of this filtering, maximum likelihood can be converged to the real position of the agent.

5.2 Kalman filtering method

Unfortunately, $\overline{h}$ made from ALV method based on the KLT tracker is not always correct homing vector due to a noisy environment. Additionally, the equal distance assumption creates deviation of measurement as a case of distribution of landmarks in surroundings. So $\overline{h}$ can be regarded as a Gaussian distribution with an average $\overline{\mu}$

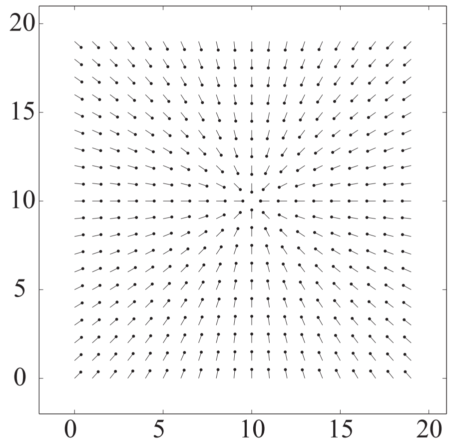


Figure 5.3: Example of ideal homing vector distribution around reference position: Reference position is (10, 10).

and standard deviation σ. If the agent observes its surroundings to estimate a homing vector, the agent can get a certain value from the Gaussian distribution as follows:

$$\overline{h} \sim N(\theta \mid \overline{\mu}, \sigma) \tag{5.9}$$

$$\overline{\mu} = \sum_{i=1}^{N} \frac{\mathbf{p_i} + \Delta \mathbf{v_i}}{\parallel \mathbf{p_i} + \Delta \mathbf{v_i} \parallel} - \sum_{i=1}^{N} \mathbf{p_i} \tag{5.10}$$

In equation 5.9, θ is sampled from the Gaussian distribution. It seems to be a difficult problem to distinguish whether the observed vector is correct. However, the specification of centralized homing vectors from a reference position helps to find an accurate homing vector by iterative checking, although measurement of the agent is changed when the agent moves to another position. Figure 5.3 shows the ideal case of the homing vector map to express specification of a ventralized homing vector. If the noise disappeared, all of the vectors are aligned toward the direction of center Therefore, if the agent observes a homing vector and noise cannot nearly effect the distortion of the homing vector, the agent can see a similar homing vector for every turn. As a result, through the history of several homing vectors, the agent can decide on the suitability of the route.

A Kalman filter based on a Bayesian approach can help to design this idea efficiently if all inputs of the Kalman filter have a Gaussian distribution because this filter can express only two parameters, such as the average and standard deviation. However, the Kalman filter estimation can be regarded as a complex model for a bio-inspired visual system to compute the whole algorithm as the following:

$$\overline{\mu_t} = A_t\mu_{t-1} + B_t u_t \tag{5.11}$$

$$\overline{\Sigma_t} = A_t\Sigma_{t-1}A_t^T + R_t \tag{5.12}$$

$$K_t = \overline{\Sigma_t}C_t^T(C_t\overline{\Sigma_t}C_t^T + Q_t)^{-1} \tag{5.13}$$

$$\mu_t = \overline{\mu_t} + K_t(z_t - C_t\overline{\mu_t}) \tag{5.14}$$

$$\Sigma_t = (I - K_tC_t)\overline{\Sigma_t} \tag{5.15}$$

where $x_t = A_tx_{t-1} + B_tu_t + \varepsilon_t$ and $z_t = C_tx_t + \delta_t$. In this case, we can simplify several parts of the equations since $A_t = C_t = I$ and $B_t = 0$ when the agent starts to move to the home location with a direct route. Even though reducing the load of Kalman filter, the designed xt as a matrix with many variables can slow the processing speed for computation of the phase of the inverse matrix.

However, if the input data for filtering have only a one-dimensional factor as an angular value, part of inverse matrix is the same as the division of a variable. In the proposed model, we can design an absolute value of an angle as only one input for the Kalman filter since deviation of the angle can only affect the state of the agent. This means that the Kalman filter can save considerable time when calculating the inverse matrix. For a one-dimensional case, updating the algorithm of the Kalman filter is as follows:

$$K = \frac{\sigma_{hist}^2}{\sigma_{hist}^2 + \sigma_{new}^2} \tag{5.16}$$

$$\overline{\mu_{update}} = \overline{\mu_{hist}} + K(\theta - \mu_{hist}) \tag{5.17}$$

$$\sigma_{update}^2 = (1 - K)\sigma_{hist}^2 \tag{5.18}$$

where K is the Kalman gain, $\overline{\mu_{update}}$ and σ_{update} are the average and standard deviation for the output of the Kalman filter, respectively. $\overline{\mu_{hist}}$ and σ_{hist} are the average and standard deviation for the history of the Kalman filter, respectively. σ_{new} is the standard deviation of the input. We set this value artificially because the standard deviation made from a real uncertainty cannot be known, but only estimated.

For iteration of the Kalman filter, the agent can find the desired homing vector efficiently. Kalman filtering can help to remove noise made from unexpected factors. If unexpected noise occurs around the measurement of the agent, the Kalman filter can control the Kalman gain as a low value to have a small effect on the Kalman average. Figure 5.4 shows the structure and an example of the proposed method.

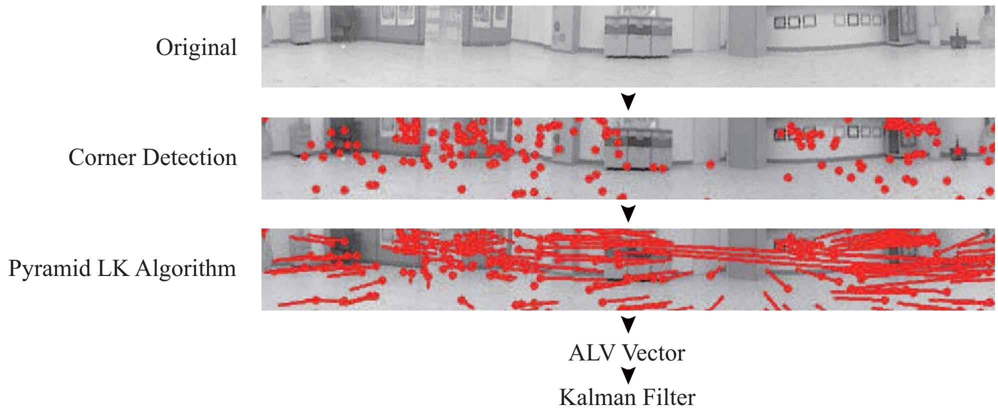


Figure 5.4: Structure and example of proposed method

5.3 Experiments and results

5.3.1 Simulations

5.3.1.1 Simulation procedure

The simulation map is a square map 20 grids wide and 20 grids high. The unit width and height can be converted to about 30cm in the real environment. This map has an outer wall with complex patterns for effective generation of the optical flow vector. Generally, the agent has only a single panoramic image, taken from the reference position. The agent can check the descriptor at intervals of moving and estimate its own position with Bayes filtering. Discrete distribution of probability can be converged by iteration of discrete filtering. To confirm the robustness of the agent localization algorithm, we control the average angular error (AAE) by adding noises in several simulations. In the general case of a local visual homing algorithm, when AAE is over a specific threshold, the performance of homing is rapidly degraded by some problems such as occlusion and deadlock with the wrong homing vector.

In this chapter, we prepare three sets of added noise at 0.4, 0.8, and 1.57 radians. The 0.4 noise set is AAE of differential flow methods in a clean environment that contains no other external noises. The 0.8 and 1.57 noise sets can be used to check performance for environments that consider external noise. In particular, in the 1.57 noise set, the agent using the existing method rarely returns to home. Figure 5.8 shows examples of the desired map and maps with some added noise to use in simulations. The agent activates the Bayes filter with the desired map information, a descriptor with local

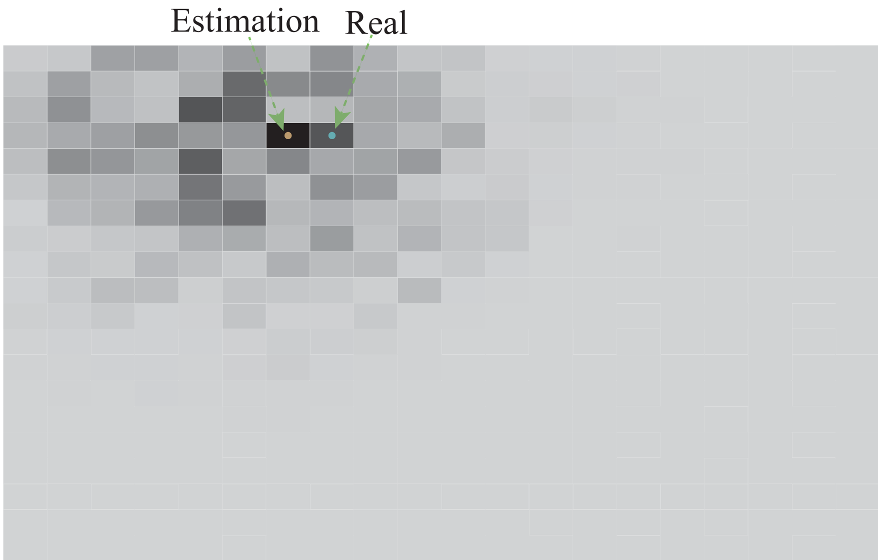


Figure 5.5: Example of agent localization on the map: Blue dot is the maximum likelihood position and red dot is the real position of agent. By only measuring its direction to the reference position, agent can find its own position information that contains distance as well as angle information from reference position.

visual homing methods as a measurement model, and an active sensing strategy to estimate its own position.

5.3.1.2 Performance of agent localization

With the three sets, we measure the performance of agent localization with different steps, such as 5, 10, 20, 50, and 100. The greater the number of steps, the more that noise with a distortion descriptor negatively affects the localization. However, the localization algorithm with the Bayes filter has a chance to converge with the clues of the measurement model and the control model.

Figure 5.5 shows an example of agent localization on the map. To measure the position estimated by the discrete Bayes filter, the maximum likelihood in a probabilistic distribution is regarded as the estimated position. The blue dot indicates the maximum likelihood position and the red dot indicates the real position of the agent in Figure 5.5. Each prior value on a discrete probabilistic distribution is sampled from a uniform distribution. The estimated position is updated by a measurement and control model. The agent does not know where it is, but it can estimate the position of the maximum likelihood value. We compare the real position and the estimated position of agent and

# of movement	Added error(rad)	Degrees error(rad)	Distance error(m)
5	0.4	0.18(0.13)	0.86(0.58)
	0.8	0.23(0.19)	0.85(0.46)
	1.57	0.45(0.29)	0.92(0.68)
10	0.4	0.23(0.18)	0.92(0.60)
	0.8	0.27(0.23)	0.95(0.64)
	1.57	0.36(0.27)	1.04(0.68)
20	0.4	0.28(0.24)	0.91(0.73)
	0.8	0.33(0.29)	1.09(0.79)
	1.57	0.42(0.42)	1.19(0.80)
50	0.4	0.20(0.17)	0.80(0.65)
	0.8	0.40(0.37)	0.99(0.72)
	1.57	0.59(0.45)	1.09(0.80)
100	0.4	0.18(0.17)	0.72(0.57)
	0.8	0.30(0.29)	0.85(0.67)
	1.57	0.52(0.48)	0.96(0.68)

Table 5.1: Results of several simulations for agent localization with noise of Gaussian distribution, average and standard deviation(in bracket): Even though noise level is high, agent estimation with Bayes filter indicates stable value. This convergence error belows average of noise described Gaussian distribution.

evaluate the performance of agent localization proposed in this paper to measure the deviation of the real position and the estimated position. Length and angular errors can be considered as criteria for measurements of deviation.

To design a noise model with a Gaussian distribution, the standard deviation of the added noise model is regarded using uncertainties of 0.4, 0.8, and 1.57. The total number of simulation practices is one hundred for each noise and movement case. The results of testing the agent localization with Gaussian noise are shown in Table 5.1. Overall, the results for each case are converged to reliable errors to estimate the agent's real positions. The rate of increase of the result deviation is relatively low when compared with the rate of increase of added angular error and the number of steps. This implies that the Bayes filtering is highly effective at supervising spatial error in agent localization.

# of movement	Added error(rad)	Degrees error(rad)	Distance error(m)
5	0.4	0.18(0.15)	0.79(0.54)
	0.8	0.27(0.20)	0.88(0.62)
	1.57	0.43(0.32)	0.84(0.60)
10	0.4	0.22(0.18)	0.95(0.62)
	0.8	0.28(0.21)	0.93(0.68)
	1.57	0.39(0.32)	0.98(0.76)
20	0.4	0.20(0.20)	1.00(0.74)
	0.8	0.34(0.32)	0.95(0.62)
	1.57	0.47(0.43)	0.91(0.67)
50	0.4	0.22(0.21)	0.79(0.68)
	0.8	0.39(0.40)	0.97(0.68)
	1.57	0.60(0.64)	1.03(0.77)
100	0.4	0.20(0.20)	0.67(0.52)
	0.8	0.33(0.28)	0.71(0.62)
	1.57	0.62(0.63)	1.01(0.68)

Table 5.2: Results of several simulations for agent localization with noise of uniform distribution, average and standard deviation(in bracket): Even though noise level is high, agent estimation with Bayes filter indicates stable value. This convergence error belows average of noise described uniform distribution.

Noise that can be represented by Gaussian distribution infers that a constant error can be expected, but unexpected noise can also exist. For example, a coherent change in the whole lighting of the environment can be regarded as expected noise, whereas moving a person or object abruptly can be considered as unexpected noise. Therefore, we test additional simulations with unexpected noise based on uniform distribution. Each uncertainty of uniform distribution is just a scaling factor in the basic range between -1 and 1. Scaling factors can also be chosen as 0.4, 0.8, and 1.57.

Table 5.2 shows the results of testing for agent localization with noise of uniform distribution. Even though the noise model is not expected by Gaussian distribution, the results are also converged as in the previous simulation. Some results even appear to show better performance than results with the noise of the Gaussian distribution. These results can provide deductive information that discrete filtering can also add robustness to various noise models.

5.3.1.3 Simulations with active sensing strategy

For improvement of the performance of agent localization, an active sensing strategy can be considered as an essential factor. The previous section covered localization that only focused on the measurement update $P(\Delta\theta_t^k|x_t^k)$ in the Bayes filter. In this section, the control update $P(x_t^k|u_t^k, x_{t-1}^k)$ is also considered by an active sensing strategy.

If the agent knows the direction of its own movement exactly, or if the agent believes strongly that it knows the direction of its own movement based on crucial evidence, active sensing can arise from any control phase. In this simulation, the agent can activate an active sensing strategy with visual or external compass information. With alignment of the agent, choosing a direction with self-energy can be one possible action. We assume that the image must be taken from an omni-directional camera at intervals of constant time to confine the control update distribution.

Figure 5.6 and 5.7 show several graphs that indicate the performance compared with a wandering strategy and an active sensing strategy of agent. The wandering strategy is one where the agent can choose a random direction of movement. The performance of the active sensing strategy is better than the performance of the wandering strategy, which shows nearly 50% error. Through biased spreading probability, the agent can avoid local optimization on a specific position with a similar measurement value to the real position more easily than it can with the wandering strategy.

5.3.1.4 Simulations with the Kalman filter

Basically, a simulation map is a square map 20 grids wide and 20 grids high. The unit width and height can be regarded as about 30cm to 50cm because, according to Vardy and Möller (Vardy and Möller, 2005), the experiment map was divided by 30cm or 50cm with width and height. This map is assumed to have an outer wall with complex patterns for effective generation of the optical flow vector. The agent gets only a single panoramic image with an omni-directional camera taken from the reference position. The agent can check its own route with the Kalman filter and decide whether this route is correct.

We design a map with a noisy environment by controlling the average angular error (AAE) by adding noise in several simulations. The AAE is one of the representative parameters for evaluating the performance of the homing vector with local visual hom-

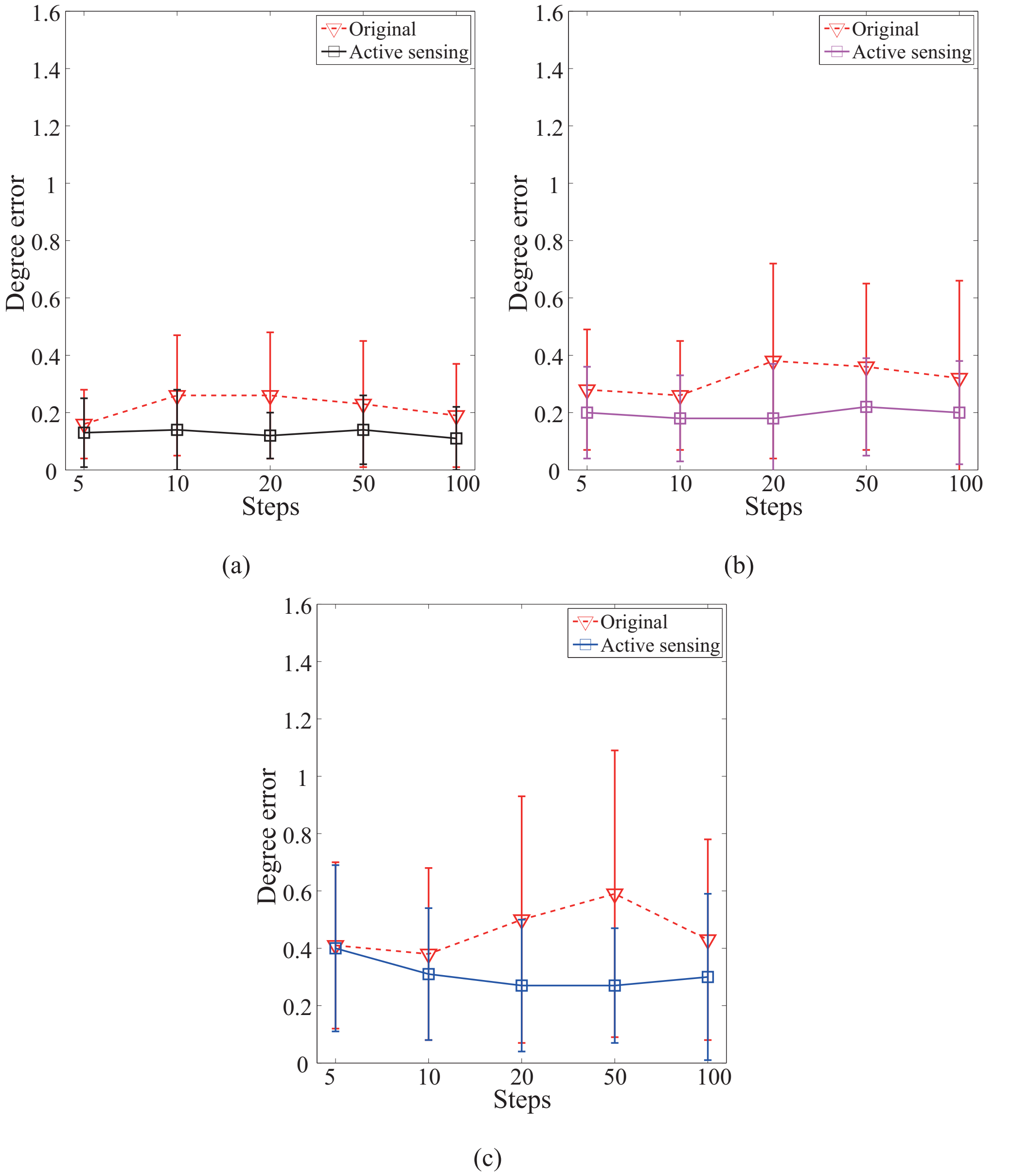


Figure 5.6: Degree error graphs to evaluate performance compared with wandering strategy and active sensing strategy of agent: (a), (b) and (c) have angular noise of 0.4, 0.8, 1.57 radian, respectively. Active sensing strategy can help to improve performance of agent localization. Some improvement can be observed when using by active sensing strategy.

ing navigation. The AAE focuses on the difference between ideal and measurement angle of homing vectors.

In this section, we prepare three maps with added noise at 0.4, 0.8, and 1.57 radians. The 0.4 noise set is AAE of differential flow methods in a clean environment, which

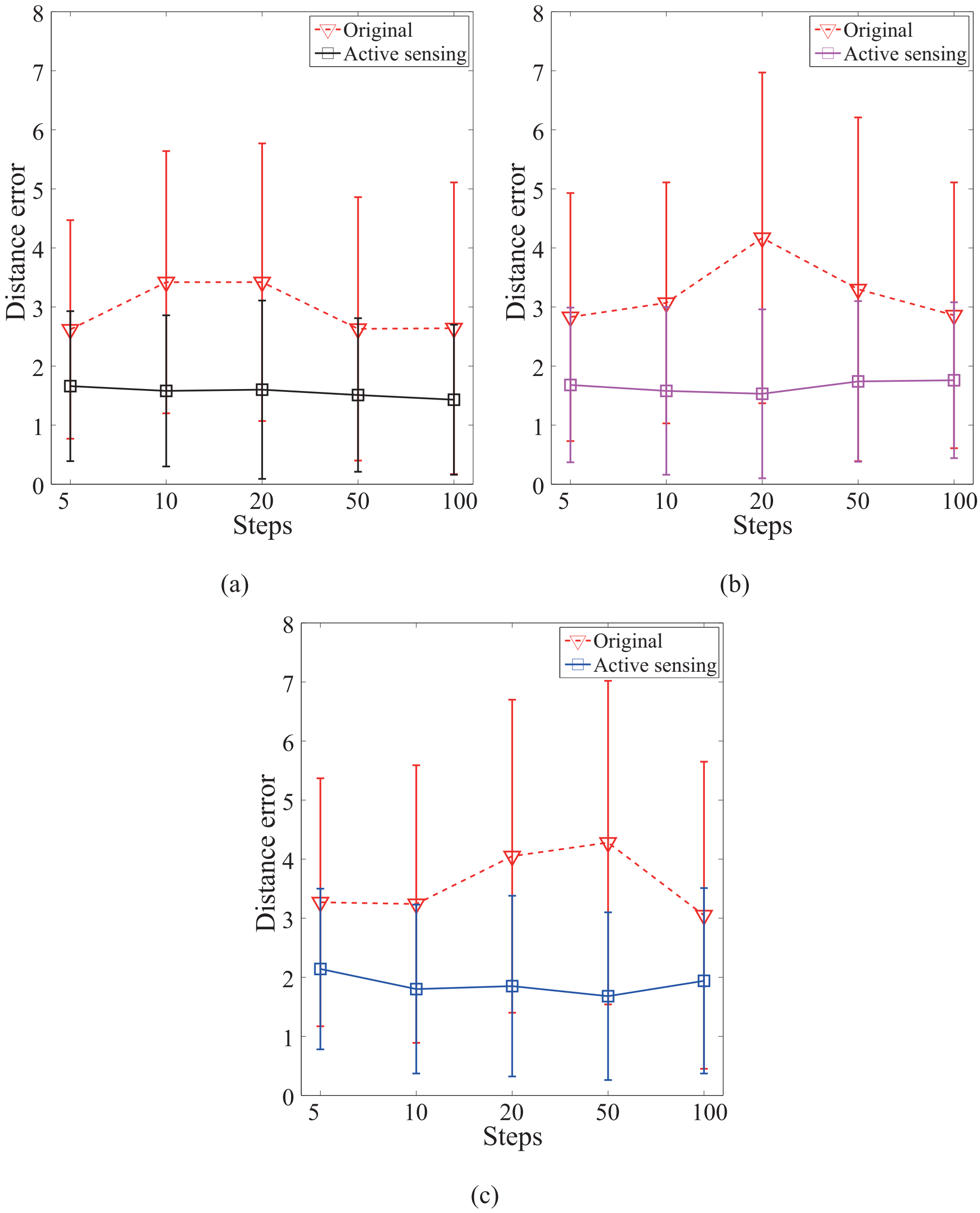


Figure 5.7: Distance error graphs to evaluate performance compared with wandering strategy and active sensing strategy of agent: (a), (b) and (c) have angular noise of 0.4, 0.8, 1.57 radian, respectively. Without directly measuring the distance from reference position, distance error increased relatively compared to the degree error but can be improved by active sensing strategy.

does not contain any other external noise. The 0.8 and 1.57 noise sets can be used to check the performance in environments that consider external noise. For the 1.57 noise set, in particular, the agent using existing method rarely returns to the home location with original method. Figure 5.8 shows examples of the desired map and maps with

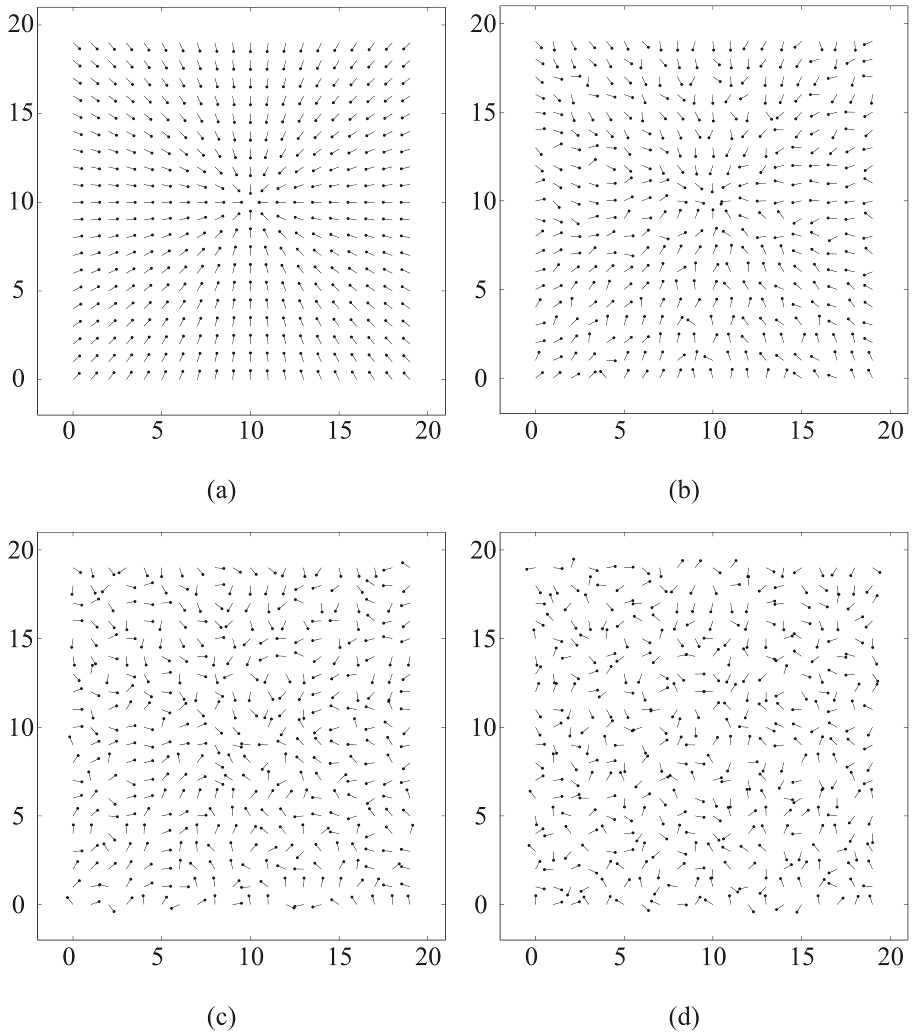


Figure 5.8: Desired map and homing vector results of sets in simulation: (a) is desired map and (b) is the case when 0.4 radians of noise is added. (c) is 0.8 and (d) is 1.57. The higher the AAE on the map, the more homing navigation and localization of agent with existing method is difficult.

some added noise to use in simulations. We compare the results of returning home with two methods: the original method and the Kalman filter method.

Table 5.3 shows results of simulation of the two methods. These results are obtained by repeating the test 100 times on the simulation map. We evaluate the performance for returning to home using the success return rate, which is an expression of the ratio for how many agents can return to the home location. In the map where a small AAE error is added, the successful return rates of both methods are similar. In contrast, the successful return rates of the two methods show broad differences in the map with the large AAE error added. This result implies that the homing method with the Kalman

Table 5.3: Results of simulation

	Noises for Gaussian distribution		
Success Return Rate	σ=22.93°	σ=45.86°	σ=90.00°
Original	0.960	0.595	0.019
Kalman filter	0.986	0.817	0.268

filter is better than the original homing method with respect to noisy environments in the simulation map.

Table 5.4: Results of simulation

	Noises for Gaussian distribution		
Success Return Rate	σ=57.32°	σ=63.06°	σ=68.79°
Original	0.347	0.223	0.149
Kalman filter	0.667	0.593	0.514

We confirmed the robustness of proposed algorithm based on the Kalman filter by testing several maps with noisy homing vectors. This additional testing used the same conditions as the previous test. Table 5.4 shows the results of the simulation for the two methods with different AAE noise. The successful return rates for the Kalman filter method were better than those of the original method.

5.3.2 Experiments

5.3.2.1 Experimental procedure

Experiments are performed on two data sets: *Original* and *Hall* from (Vardy and Möller, 2005). These are image data sets without any pre-processing to improve extraction features or descriptors. Examples of the image data sets are shown in figure 5.9. Each image data set has a complex background that is covered with wide open space on the outside. In this case, extraction of features and estimating relative movement from features can be difficult with existing methods. Even if feature extraction is successful, the high computational load interrupts real-time processing.

In these experiments, the agent uses an agent localization algorithm with active sensing strategy. Instead of the noise models with Gaussian or uniform distribution, real

(a)

(b)

Figure 5.9: Examples of image sets: (a) is *Original* and (b) is *Hall*. These image sets are released at *http://www.ti.uni-bielefeld.de/html/research/avardy*.

# of movement	Degrees error(rad)	Distance error(m)
5	0.12(0.16)	0.32(0.29)
10	0.09(0.15)	0.18(0.22)
20	0.10(0.15)	0.25(0.34)
50	0.09(0.11)	0.28(0.30)
100	0.10(0.10)	0.33(0.35)

Table 5.5: Results of experiment for agent localization with *Original* image sets: degree and distance error can be converged by proposed algorithm.

descriptors measured from each position are used. Descriptors are made from differential flow methods using ALV based on a KLT algorithm. The distortion of the omnidirectional camera and the noise on the raw images are not calibrated to help to correct result and to reduce processing time. The rest of conditions are almost the same as in the simulation testing.

Essentially, the agent uses a visual compass algorithm to align itself in the initial direction, instead of using external compass. If the agent can use an external compass in the environment, the processing speed is faster than for a case using a visual compass algorithm. Although additional error is incurred from the alignment of the agent, Bayes filtering can reduce erroneous angle problems with the probabilistic approach.

# of movement	Degrees error(rad)	Distance error(m)
5	0.19(0.51)	0.29(0.28)
10	0.11(0.25)	0.21(0.28)
20	0.16(0.17)	0.27(0.29)
50	0.15(0.12)	0.34(0.31)
100	0.20(0.16)	0.44(0.38)

Table 5.6: Results of experiment for agent localization with *Hall* image sets: : degree and distance error can be converged by proposed algorithm.

5.3.2.2 Results of experiments

Table 5.5 and 5.6 show the results of experiments on agent localization with the Original and Hall image sets, respectively. In real cases, the performance of agent localization can be converged stably with differential flow methods and active sensing strategy, despite the complex scene that is located irregularly outside.

Precise results are obtained by capturing the image used with differential flow methods at constant intervals. Therefore, in the *Original* set, since each grid has intervals of about 30cm in the histogram filter, the agent has the assumption that step size is constant at about 30cm.

In the *Hall* set, the agent has intervals of about 50cm in the histogram filter. If a localization map is designed as a particle filter, these intervals can still be applied for unit movement of each particle in the map. If the distance of the control update can be changed by several factors, this can be a good alternative method for a control model that has Gaussian distribution for distance as well as for direction of movement.

When the agent activates an active sensing strategy, the choice of direction of routes is not important. The mere belief in the movement direction is sufficiently reliable. If the agent strays out of range on the way to the active sensing strategy, it can change its direction of movement for using compass information. Even though the agent cannot know what is out of range for the reference image, it can estimate the limitation of the difference between the current image and the reference image by using the KLT algorithm. However, in some cases, the agent must be moved to an area far from the reference position. In the next section, we will consider the limitations of this algorithm and ways to overcome limitation problems.

Added error(rad)	Avg. of error distance (m)	Std. of error distance
0.4	2.82	1.76
0.8	3.29	2.00
1.57	3.00	1.99

Table 5.7: Results of simulation for agent localization with large map: Agent uses several snapshot images taken from reference positions to estimate its own position. Results did not consist of degree error because there were more than one snapshot images are not single in this case. So we replace error descriptions to calculate distance between real position and estimated position.

5.3.2.3 Localization simulation of wide map

In this section, we suggest two conditions for choosing a suitable reference image. The first condition is to match the rate of how many corners, after Shi-Tomashi corner detection, can be made into optical flow vectors using the pyramid LK algorithm. A low matching rate means that the difference between the two images is high. If an image taken from the current position is similar to a reference image, the agent can guess that the two images are taken from adjacent locations. The second condition is the total length for the sum of the optical flow vectors. This concept is different from ALV because the definition of length does not contain any sign. If the length for the sum of optical flow vectors is long, the two images can be guessed as being obtained from distant locations.

Table 5.7 shows the results of performance for long range navigation with the multireference position approach. In this simulation, we assume that the agent has 100 steps constantly for each test. Even though the total noise is greater than for the case of a single reference position approach, the result is still sufficiently reliable to apply to a real system. This idea can be applied to guide a robot in a museum where the agent knows location information or has images taken from several positions in advance. To avoid noise, image sets can be taken from the ceiling instead of from the floor.

5.3.2.4 Real time tracking of the original method

We use the laboratory to evaluate the proposed algorithm. In the laboratory, the agent moves on a map with dimensions of about 3.5m $\times$ 3.5m. Figure 5.10 shows the en-

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

Figure 5.10: Scenes of experimental arena: (a) is front, (b) is left, (c) is right and (d) is behind.

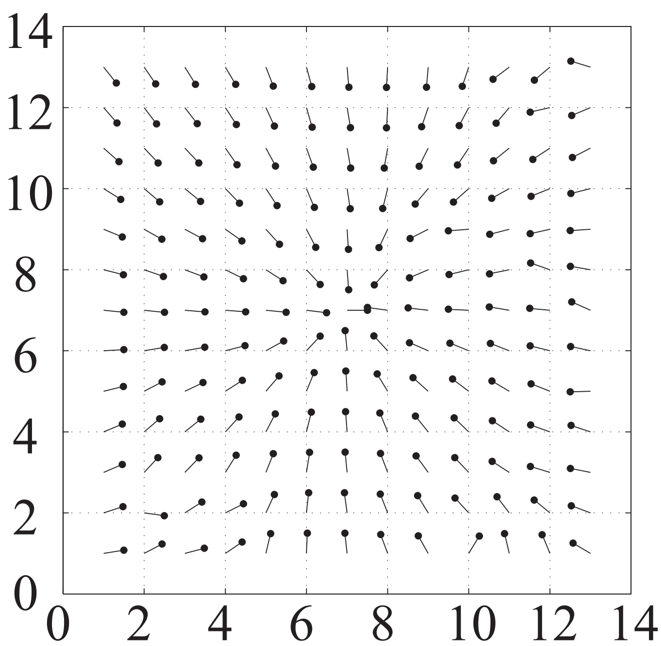


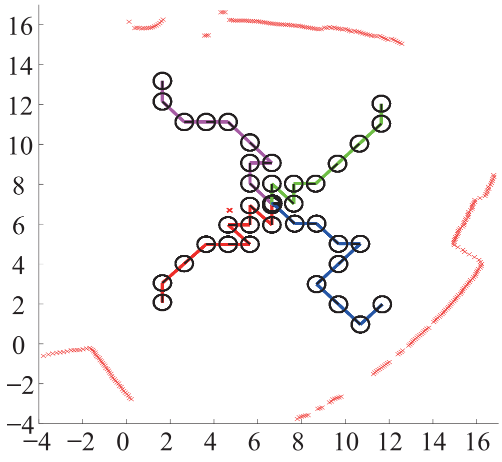
Figure 5.11: Homing vector map of real case in laboratory without any noise

vironment for an experiment with the agent, where the agent finds a homing direction within a complex environment. The agent takes 169 pictures and calculates an ALV vector as shown in Figure 5.11 without any noise excepting for basic noises occurring in the environment. Objects which are general laboratory goods on the map are allocated to the outside of the map. These are clues for making corners and landmarks for the agent to recognize the homing direction. Distances from the agent to objects are differences for each object, but based on the equal distance assumption, the agent regards all distances to be the same. Measurement data are obtained from the omni-directional camera and are transformed as a panoramic image data. The KLT and ALV algorithm can then be applied to panoramic image to estimate the homing direction. Pioneer has one marker above the omni-directional camera for localizing a global camera attached to the ceiling in the laboratory. We can check the spatial state of the agent with this global camera.

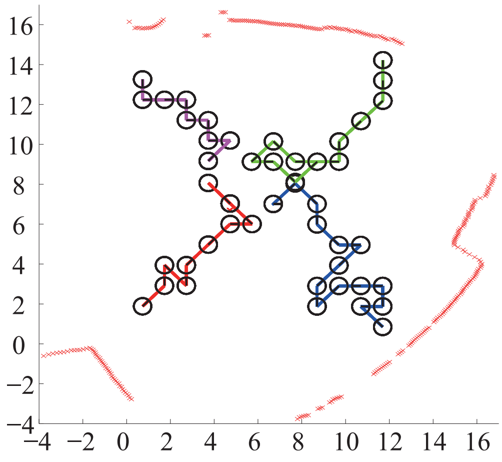
To detect the homing position, the agent checks the optical flow vector sizes and detects corners. If the resulting values are over specific thresholds from measurement updates, the current position is regarded as the home for the agent. A histogram equalization process helps to add robustness to the algorithms for visual images when the light changes. Distortion of the intensity of light can affect the creation of optical flow vectors.

We use a Pioneer robot to confirm the performance of the proposed algorithm. Before testing this algorithm, the original algorithm is tested on the map. In a real case with the original method, the agent cannot find an accurate route to return home because of external noise due to reflection of light from floor, movement of unexpected objects, camera calibration, and movement of the Pioneer.

These results of homing navigation of the agent with original method are shown in Figure 5.12. which indicates the track of the Pioneer with the original method. The objective of the agent is to move to the center position from a random position. In Figure 5.12, the agent in some cases has difficulty in returning to home because of external noise. The agent wanders on the map with wrong visual data and diverse spatial states. Therefore, in this case, a probabilistic filter could be needed to remove noise from the visual information in order to estimate an accurate homing vector.



(a)



(b)

Figure 5.12: Examples of homing navigation with original method in noisy map: (a) has noises with Gaussian distiribution of σ=45.86° (b) has noises with Gaussian distiribution of σ=68.79°

5.3.2.5 Real time tracking of the proposed algorithm

In a Kalman filter, minor information with noise is neglected or rejected because general noise does not appear steadily. Even though bias information appears steadily, this is not a good information and cannot be detected by the Kalman filter. Therefore, the proposed algorithm with the Kalman filter is effective at finding a homing route for the agent.

Figure 5.13 shows the results for the proposed algorithm with the Kalman filter. These results imply that proposed method is robust for a noisy environment. The agent can find a route to return home after major information from correct visual clues are caught by the Kalman filter and the agent can catch major information in the early phase of all cases. However, the Kalman filter can cause side effects that result in the wrong homing of the agent when visual cues provide a major homing direction with constant wrong information.

We compared the original and Kalman filter methods by designing a noisy map based on the real homing direction map. First, we estimate the noise level of a real case to check the distortion of the homing direction in a sampling case in which the total number is 40 times from different positions. We find that the noise level of the real case is about 42°. Second, a visual homing map that is noise free can have noise added with a Gaussian distribution of standard deviation, which is 42°.

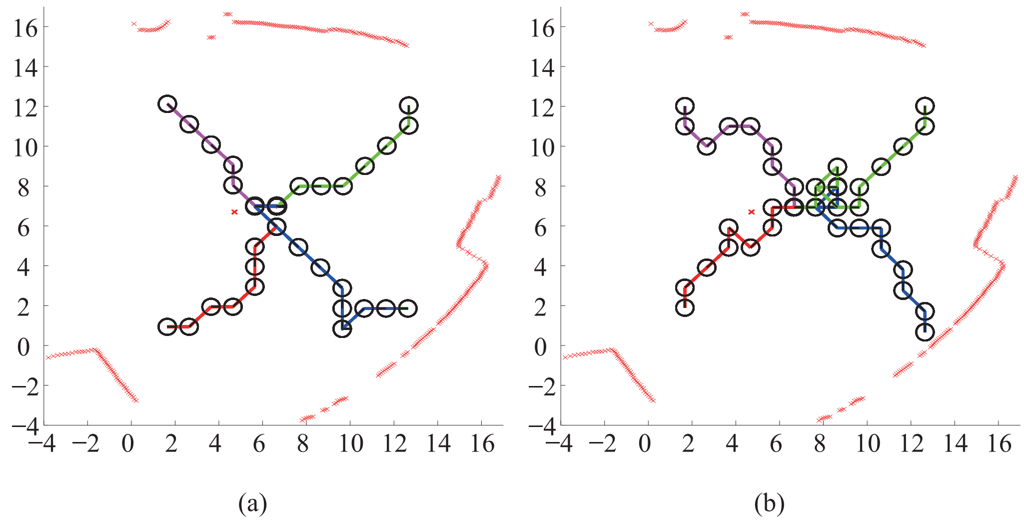


Figure 5.13: Examples of homing navigation with Kalman filter method in noisy map: (a) has noises with Gaussian distiribution of σ=45.86° (b) has noises with Gaussian distiribution of σ=68.79°

The results of testing the real case are shown in Table 5.8. These results came from 1000 tests for the agent to move from each random position to the home position. Even though the step number with the Kalman filter is greater than the step number for the original method, the successful return rate with the Kalman filter is better than that of the original method. The agent with the Kalman filter initially wanders on the map because of the phase to find a major homing vector. Therefore, the total step number can be increased.

Consequently, tables 5.3, 5.4 for simulation and 5.8 for the real experiment show that the Kalman filter method is superior to the original method in an environment with complex noises.

5.3.3 Limitation of real cases

The proposed model has several limitations when applied to real cases. First, a change in global illumination will cause matching with the optical flow algorithm to fail. This problem comes about for two main causes: being out of range for detection and automatic calibration of the camera. Being out of range for detection is not really a change in global illumination but the agent can consider it to be. This model has a certain range for detecting matching to estimate the homing direction. However, if the agent moves

Table 5.8: Results of Experiment

	Noises for Gaussian distribution σ=42°		
	Success Return Rate	Step Avg.	Step Std.
Original	0.764	10.64	5.35
Kalman filter	0.844	11.96	3.96
	Noises for Gaussian distribution σ=45.86°		
	Success Return Rate	Step Avg.	Step Std.
Original	0.685	11.88	5.78
Kalman filter	0.770	13.04	4.27
	Noises for Gaussian distribution σ=68.79°		
	Success Return Rate	Step Avg.	Step Std.
Original	0.282	16.90	5.08
Kalman filter	0.451	16.80	3.93

out of range for detection, it can confuse its current state compared with the reference state and cannot match corners with the optical flow algorithm. Automatic calibration problem is a hardware issue. When the camera turns on, it can automatically control the intensity of light to adapt to the environment to create a condition where the camera can take a picture. A change in intensity creates an error for matching with the optical flow algorithm and results in failure to estimate the homing vector.

Second, the initial failure of estimation creates difficulty in fixing the homing direction. The concept of a Kalman filter refers to the history of states and makes a correct weight for each parameter to estimate the next average and standard deviation. However, if the agent encounters erroneous information, the agent believes the wrong history of states and makes a wrong weight for each parameter. This incurs deadlock problems for iterative movement in the same position.

5.4 Comparison for the proposed algorithms

In this section, we introduce a comparison for the proposed algorithm to estimate the homing direction. We talk about the original method, the Kalman filter method, and the desired map method. The testing method is the same as presented in the previous section.

Table 5.9: Results of Simulation

	Noises for Gaussian distribution		
Success Return Rate	σ=22.93°	σ=45.86°	σ=90.00°
Original	0.963	0.594	0.020
Kalman filter	0.987	0.827	0.272
Desired map	1.000	1.000	0.770

The desired map method can make the greatest performance compared with any other method. This method is especially robust for diverse noise distribution as well as Gaussian distribution of noise because this consists of non-parametric filters such as a histogram filter or particle filter. The Kalman filter has the limitation that this method can only reduce noise when the noise model is Gaussian. Table 5.9 shows the results for each method.

The Kalman filter method has an advantage of processing speed since its design consists of two factors: essentially just the average and standard deviation, but the desired map must have all of the probabilistic information for spatial states on the range of map. This can be regarded as inefficient parts for computational load and memory compared with other methods. In the real case, if the range of detectable positions is very large, computation will be increased exponentially for the desired map case.

Table 5.10: Comparison for proposed methods

	Original	Kalman filter	Desired map
Performance	−	+	+++
Processing speed	+++	++	0
Implementation	++	0	−
Memory efficiency	+++	++	0

Table 5.10 shows a comparison of the proposed algorithms. + indicates good performance, 0 is moderate performance and ` is bad performance. Depending on the conditions, each visual model can be applied to returning to home.

5.5 Summary of Chapter 5

In this chapter, we introduce a probabilistic approach with local visual navigation to correct the measurement model for uncertainty of the environment. Cases of the probabilistic model can be regarded as two types. First, each measurement part can be corrected using the desired map. According to the basic specification of the local visual homing navigation with differential flow methods, a centralized homing direction to home can be regarded as a desired map. This microscopic aspect with the desired map can be controlled by a nonparametric Bayesian filter such as a histogram filter or a particle filter.

Second, the tendency for exploration can also be applied to the Kalman filter to correct the homing direction. Even though all conditions of the agent with homing navigation have a complex structure, when the agent finds an accurate route home, the spatial state model of the agent can be regarded as simple cases of the Kalman filter. This macroscopic model with a Kalman filter is faster than the method with the desired map.

From these probabilistic models, the agent can control its own movement with robustness for external noise. The localization problem can therefore be solved in a local visual homing navigation and the performance of homing direction can be upgraded by preventing distortion of the measurement model.

Chapter 6

Conclusions

Visual navigation based on optical flow is one of the bio-inspired approaches to engineering. To stand out advantages of bio-inspired approaches against existing engineering methods, efficiency of computation can be regarded as essential factor to estimate performance of proposed model. Without complex image processing such as pattern recognition about high dimensional data, the bio-inspired measurement model can suggest new approaches such as ALV, which uses basic concepts where the image changes monotonously when an agent with an omni-directional camera moves from home to a certain position. This model can make moderate performance of visual navigation in spite of a lack of visual information for the environment.

The proposed models in this paper add several contributions to robot engineering. First, the accumulative visual navigation can imply the possibility of path integration with only visual information. Second, the measurement model of ALV can make landmarks automatically in a random environment even though the surroundings around the agent consist of complex objects that are difficult to distinguish as landmarks. Third, the introduction of a low dimensional Bayesian approach using a parametric or non-parametric filter such as a Kalman filter and a particle filter can help to add robustness to the performance for visual navigation of an agent with a low computational load. Lastly, localization without a pre-searching phase for recognizing or estimating the spatial state is feasible with only one omni-directional image taken from the reference position through the Bayesian approach with a desired map inspired by place cells and grid cells in biology.

Consequently, the agent can return to home and can recognize its own position with only a single omni-directional camera and two images taken from the current location and reference location. A pre-searching phase is not necessary for the agent with the proposed algorithm because the whole map information is already given in the reference image. This algorithm with the Bayes filter uses a simple measurement model and shows active sensing strategy and is robust against noise and efficient in its computational cost. Even if navigation method of the ant cannot be investigated exactly, we can confirm that bio-inspired model can help to agent localization in engineering.

6.1 Flow methods to estimate the homing vector

Through several tests, we can confirm that a combination of the sector approach and the differential flow method for visual homing navigation contributes to saving time and reduces the numbers of corners and sectors. In this section, we will talk about further issues for homing navigation. Even if the block matching method has a better homing performance than the differential flow method based on the KLT algorithm, a time problem arises from the complexity of the search area used in block matching. As the search range of block matching increases, time is also increased by $O(n^2 n_x n_y)$. Vardy and Möller (Vardy and Möller, 2005)suggest the IntMatch method and the GradMatch method to reduce the complexity of computation, even though neither of these methods outperforms BlockMatch. Similarly, we also give a sector based model by removing trivial features through setting a threshold for the eigenvalue for autocorrelation matrix λ. Reducing the number of features decreases the coefficient of n^2 in $O(nN + n^2)$. A low angle resolution of image can also maintain the homing performance of the agent. This means that the image remaining after the process of reducing its size from the original capture image can be regarded as moderating the sample in the estimating direction.

Another issue is long-range visual homing. If the limitation of the search or orientation range used for accurate calculation of the value must be confined, then only one snapshot cannot applied to long-range visual homing using this model. Vardy suggests a multi-snapshot model to navigate a wide environment when using bio-inspired navigation based on intensity information (Vardy, 2006). Automatic captures are activated with several conditions that are considered based on two parameters: the difference in angle and distance. With multi-snapshots, the agent can be navigated over long distances regardless of rapidly changing environmental patterns.

Further work is needed to solve localization problems when the agent encounters a kidnapped situation; these problems can arise from unwanted movement due to an external force or temporary breakdown. We have several choices for estimating the position of an agent that directly uses particle filters based on an intensity measurement model or that adds an additive filter update to the phase of measurement update based on a bio-inspired approach from neural networks (Sunderhauf and Protzel, 2010).

6.2 Probabilistic approach of bio-inspired navigation

In this paper, the agent can recognize its own position with only a single omni-directional camera and two images taken from the current location and a reference location. Moreover, a presearching phase is not necessary for the agent with the proposed algorithm because the whole map information is already given from the reference image. This algorithm with a Bayes filter using a simple measurement model and active sensing strategy is robust in the presence of noise and efficient in its computational cost. Even if the navigation method of the ant cannot be investigated precisely, we can confirm that a bio-inspired model can help in agent localization in engineering.

In further work, we will study efficient movement strategy in a noisy environment without the desired map. Unlike mammals, insects do not have place cells and grid cells in their brain. Nevertheless, they can choose their route robustly in a complex environment. Reducing the error of the proposed algorithm in long range navigation can be considered another issue. To solve this problem, measurement criteria need to be stricter than used for the presently proposed algorithm.

Visual homing navigation with a Bayesian approach can help to remove unexpected noise in reality. Our approach is based on optical flow and the appropriate landmark vectors are extracted from the optical flow. A set of landmark vectors determine the homing direction. The approach experiences noise-sensitive estimation and the Bayesian approach in a history of estimation can improve the performance of returning home. One dimensional Kalman filtering is an effective method for visual homing navigation that estimates an accurate homing direction with a relatively small computing time.

6.3 Future works

6.3.1 Landmark matching in simple environment

The results of this paper show that all environments consist of complex objects.I f an agent applies the proposed algorithm to a simple environment, the performance of returning home can be degraded due to any reduction in matching points in the surroundings of the agent. A combination of ALV with a cluster matching algorithm and an optical flow algorithm is a reasonable way to maintain performance. However, distinguishing environment conditions and making criteria to judge the complexity of the environment are important. One of the probable solutions is to check the number of corners in the KLT. If the KLT makes a small number of corners compared with a certain threshold value, the environment around the agent can be regarded as a simple background. Nevertheless, this method would not distinguish erroneous cases because a case that involves a matching failure could also show a small number of corners in the KLT.

In another approach, the agent could recognize patterns of objects instead of changes in the image. However, occasionally, this can incur a heavy load on computation and memory. To prevent this problem, we have to determine which patterns are robust for a changing environment and for adding noise, so a neural network algorithm with small nodes to distinguish patterns could be a good choice. Through the learning phase, a neural network can choose robust patterns for estimating a spatial state. Hebbian learning and a back-propagation algorithm as a learning algorithm can be used to design the model.

After recognizing patterns, a cam-shift algorithm and a particle filter algorithm can track the clustering of landmarks in the environment. Compared with a mean-shift algorithm, these algorithms can be converged globally in random surroundings of the agent. Even if occlusion of landmarks is encountered, estimation of landmarks can help to prevent degradation of performance in calculating homing vectors. However, this opens up a new question because continuous images taken from a camera are needed for a snapshot assumption from a bio-inspired model.

6.3.2 Robust feature detection

The optical flow algorithm is powerful enough to recognize a changing image but this is not a precise image matching because the KLT algorithm does not always guarantee matching success between corners to corners for each image. Image matching with a scalar invariant feature transform (SIFT) algorithm seems to be a good alternative choice to increase the matching probability. Nevertheless, this is not a simple problem, since the matching algorithm has to contain a changing distribution from FOC and FOE as well as image matching.

Another issue of robust feature detection is removal of noise from the taken image and matching results. Performance can be upgraded to find the homing direction as descriptor by using maximally stable extreme regions (MSER) in the preprocessing phase of image matching. The MSER method is similar to a watershed algorithm for distinguishing segmentation but the precision of performance is better than the performance of a watershed algorithm.

Random sample consensus (RANSAC) can also be a good choice for removing noise to compare with other results in the measurement phase. RANSAC can discard minor factors from major factors in clustering. In this case, a post-processing phase after creation of flow vectors can be used to distinguish noisy vectors. We can design a certain threshold value to improve the performance of filtering by testing the algorithm for several cases. RANSAC is usually faster than other algorithms and can be used in nearly real-time when data collection is incomplete. With several methods, selection of a robust feature for a noisy environment means that the performance for returning to home can be upgraded by reducing the error.

6.3.3 Application of a wide map

Even though differential flow methods have a robust measurement model for distance from the reference position, long-range navigation is still a critical problem for the algorithm proposed in this paper. The main cause of this problem is a lack of information about the panoramic image taken from the omni-directional camera at the reference position. The further the agent moves away from the reference position, the greater the probability that matching will decrease for the snapshot image and another image taken from a certain position. To solve this limitation, a multi-reference position approach is needed where the agent has several snapshot images taken from several reference positions.

To apply a multi-reference position approach, an important additional rule is to decide which reference image can help with agent localization. Generally, the closest located image can be chosen as a reference image for the agent, but, in this case, the agent cannot know the distance from its current position and the reference image located at a certain position. Therefore, the agent uses a different method to find what image is taken from the nearest position.

In biology, if a desert ant has a mapping system that is similar to a desired map, the ant can find its own position information based on only two sightings that are taken from its current position and its nest. The compound eye of the ant can be regarded as a system for creating an optical flow as a visual sensory system and celestial information, such as the sun, moon, and stars, can help to orientate the ant. A specific track of the ant, like a circle, can be considered to represent an active sensing strategy to help fast convergence of the ant localization. However, these are still open questions regarding what the ant uses in real cases and why the ant moves rotationally around its nest.

6.3.4 Application of SLAM

Unlike SLAM, since the proposed model can explore a map using an omni-directional image taken from the reference position instead of map information, the agent can move with global localization from its initial position using just a localization algorithm. However, SLAM is one of the popular issues in robotics, and automatic mapping is an important factor of intelligent robotics, so the application of SLAM with bio-inspired navigation is meaningful work in engineering fields.

The proposed model with a desired map has advantages in computational efficiency and robustness of the environment that lend it to a SLAM design. Drawing part of the map can be regarded as adding an automatic capture phase in the proposed localization with several sensors. The main problem with application of SLAM to a bio-inspired approach is the increasing exponential complexity when the agent has many waypoints to draw the map and extend the range of navigation because a folded range with several waypoints must be compared with multiple images taken from each waypoint position. To avoid this problem, we can consider the factorization property of the measurement phase. In an optional case, we can also consider a Markov chain Monte Carlo (MCMC) approach to estimate the agent's own position and draw a specific map.

When the agent is started, it has only one omni-directional image taken at the reference position. The agent can collect map information with its omni-camera and with other sensors moving around from the reference position. If the agent goes out of range of the detectable homing vector as a descriptor of localization, the agent takes a new picture from its current position. In this manner, the agent can accumulate visual information for the whole map. To localize an agent for a new measurement model, the agent can compare several images taken from several reference positions to estimate its current position. In this case, the agent can use a non-parametric filter with a sectorization property or MCMC to fix its estimation of its current position and reference positions where it took the pictures.

Bibliography

Adorni, G., Cagnoni, S., Enderle, S., Kraetzschmar, G., Mordonini, M., Plagge, M., Ritter, M., Sablatnög, S., and Zell, A. (2001). Vision-based localization for mobile robots. *Robotics and Autonomous Systems*, 36(2):103–119.

Angulo, C. and Godo, L. (2007). Using the average landmark vector method for robot homing. *Artificial Intelligence Research and Development*, 163:331.

Argyros, A., Bekris, K., and Orphanoudakis, S. (2001). Robot homing based on corner tracking in a sequence of panoramic images. In *In Proceedings of the 2001 IEEE Computer Vision and Pattern Recognition*, volume 2, pages II–3. IEEE.

Argyros, A., Bekris, K., Orphanoudakis, S., and Kavraki, L. (2005). Robot homing by exploiting panoramic vision. *Autonomous Robots*, 19(1):7–25.

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):1–16.

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.

Baird, E., Srinivasan, M., Zhang, S., Lamont, R., and Cowling, A. (2006). Visual control of flight speed and height in the honeybee. *Simulation of Adaptive Behavior 2006: From Animals to Animats 9*, pages 40–51.

Baker, S. and Matthews, I. (2004). Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255.

Barron, J., Fleet, D., and Beauchemin, S. (1994). Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77.

Beauchemin, S. and Barron, J. (1995). The computation of optical flow. *ACM Computing Surveys (CSUR)*, 27(3):433–466.

Bellotto, N., Burn, K., Fletcher, E., and Wermter, S. (2008). Appearance-based localization for mobile robots using digital zoom and visual compass. *Robotics and Autonomous Systems*, 56(2):143–156.

Binding, D. and Labrosse, F. (2006). Visual local navigation using warped panoramic images. In *In Proceedings of Towards Autonomous Robotic Systems. University of Surrey, Guildford, UK*, pages 19–26. Citeseer.

Bolles, R. and Fischler, M. (1981). A ransac-based approach to model fitting and its application to finding cylinders in range data. In *In Proceedings of Seventh International Joint Conference on Artificial Intelligence*, pages 637–643.

Bouguet, J. (2001). Pyramidal implementation of the affine lucas kanade feature tracker—description of the algorithm. Technical report, Intel Corporation, Microprocessor Research Labs.

Bradski, G. (1998). Computer vision face tracking for use in a perceptual user interface. *Interface*, 2(2).

Brox, T., Bregler, C., and Malik, J. (2009). Large displacement optical flow. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 41–48. IEEE.

Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. *Computer Vision-ECCV 2004*, pages 25–36.

Burke, A. and Vardy, A. (2006). Visual compass methods for robot navigation. In *Seventeenth Annual Newfoundland Electrical and Computer Engineering Conference*. Citeseer.

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.

Cha, Y.S. Baek, S. and Kim, D. (2013a). Visual homing navigation with discretizing step method based on optical flow (being prepared). *Adaptive Behavior*.

Cha, Y. and Kim, D. (2012a). Homing navigation based on path integration with optical flow. *Journal of Institute of Control, Robotics and Systems (In Korean)*, 18(2):94–102.

Cha, Y. and Kim, D. (2012b). Omni-directional image matching for homing navigation based on optical flow algorithm. In *2012 IEEE International Conference on Control, Automation and Systems*, pages 1446–1451.

Cha, Y. and Kim, D. (2013b). Agent localization based on active sensing strategy with bio-inspired measurement model (being prepared). *Adaptive Behavior*.

Cha, Y. and Kim, D. (2013c). Bio-inspired application of local visual homing navigation with bayesian approach (submitted). In *The 2013 IEEE International Conference on Robotics and Automation 2013*.

Cha, Y. and Kim, D. (2013d). Combination of sector apporach and differential flow method for visual homing navigation (being prepared). *Adaptive Behavior*.

Collett, T. and Lehrer, M. (1993). Looking and learning: a spatial pattern in the orientation flight of the wasp vespula vulgaris. *In Proceedings of the Royal Society of London. Series B: Biological Sciences*, 252(1334):129–134.

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619.

David, C. (1979). Height control by free-flying drosophila. *Physiological Entomology*, 4(3):209–216.

Fennema, C. and Thompson, W. (1979). Velocity determination in scenes containing several moving objects. *Computer Graphics and Image Processing*, 9(4):301–315.

Fischler, M. and Bolles, R. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Fox, D., Burgard, W., Thrun, S., and Cremers, A. (1998). Position estimation for mobile robots in dynamic environments. In *Proceedings of the National Conference on Artificial Intelligence*, pages 983–988. John Wiley & Sons.

Fox, D., Hightower, J., Liao, L., Schulz, D., and Borriello, G. (2003). Bayesian filtering for location estimation. *2003 IEEE Coference on Pervasive Computing*, 2(3):24–33.

Franceschini, N. (2004). Visual guidance based on optic flow: a biorobotic approach. *Journal of Physiology-Paris*, 98(1-3):281–292.

Franceschini, N., Ruffier, F., and Serres, J. (2007). A bio-inspired flying robot sheds light on insect piloting abilities. *Current Biology*, 17(4):329–335.

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.

Franz, M., Stürzl, W., Hübner, W., and Mallot, H. (2008). A robot system for biomimetic navigation–from snapshots to metric embeddings of view graphs. *Robotics and Cognitive Approaches to Spatial Mapping*, pages 297–314.

Frier, H., Edwards, E., Smith, C., Neale, S., and Collett, T. (1996). Magnetic compass cues and visual pattern learning in honeybees. *Journal of Experimental Biology*, 199(6):1353–1361.

Fry, S., Rohrseitz, N., Straw, A., and Dickinson, M. (2009). Visual control of flight speed in drosophila melanogaster. *Journal of Experimental Biology*, 212(8):1120.

Gaussier, P., Joulain, C., Banquet, J., Leprêtre, S., and Revel, A. (2000). The visual homing problem: an example of robotics/biology cross fertilization. *Robotics and Autonomous Systems*, 30(1):155–180.

Gibson, J. (1950). The perception of the visual world. *Riverside Press, Cambridge*.

Goedemé, T., Nuttin, M., Tuytelaars, T., and Van Gool, L. (2004). Vision based intelligent wheel chair control: the role of vision and inertial sensing in topological navigation. *Journal of Robotic Systems*, 21(2):85–94.

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*, pages 1806–1811. IEEE.

Gourichon, S., Meyer, J., and Pirim, P. (2002). Using coloured snapshots for short-range guidance in mobile robots. *International Journal of Robotics and Automation*, 17(4):154–162.

Graham, P., Philippides, A., and Baddeley, B. (2010). Animal cognition: multi-modal interactions in ant learning. *Current Biology*, 20(15):R639–R640.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Alvey Vision Conference*, volume 15, page 50. Manchester, UK.

Harris, R., Graham, P., and Collett, T. (2007). Visual cues for the retrieval of landmark memories by navigating wood ants. *Current Biology*, 17(2):93–102.

Hatzitheodorou, M., Karabassi, E., Papaioannou, G., Boehm, A., and Theoharis, T. (2000). Stereo matching using optic flow. *Real-Time Imaging*, 6(4):251–266.

Homberg, U. (2004). In search of the sky compass in the insect brain. *Naturwissenschaften*, 91(5):199–208.

Horn, B. and Schunck, B. (1981). Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203.

Jain, J. and Jain, A. (1981). Displacement measurement and its application in interframe image coding. *IEEE Transactions on Communications*, 29(12):1799–1808.

Kennedy, J. (1951). The migration of the desert locust (schistocerca gregaria forsk.). i. the behaviour of swarms. ii. a theory of long-range migrations. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, pages 163–290.

Kirchner, W. and Srinivasan, M. (1989). Freely flying honeybees use image motion to estimate object distance. *Naturwissenschaften*, 76(6):281–282.

Koenderink, J. and Doorn, A. (1987). Facts on optic flow. *Biological Cybernetics*, 56(4):247–254.

Kuglin, C. and Hines, D. (1975). The phase correlation image alignment method. In *In Proceedings of International Conference Cybernetics Society*, pages 163–165.

Labhart, T. and Meyer, E. (2002). Neural mechanisms in insect navigation: polarization compass and odometer. *Current Opinion in Neurobiology*, 12(6):707–714.

Labrosse, F. (2004). Visual compass. *In Proceedings of Towards Autonomous Robotic Systems, University of Essex, Colchester, UK*, pages 85–92.

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., Möller, 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.

Lindemann, J., Kern, R., van Hateren, J., Ritter, H., and Egelhaaf, M. (2005). On the computations analyzing natural optic flow: quantitative model analysis of the blowfly motion vision pathway. *The Journal of Neuroscience*, 25(27):6435–6448.

Lucas, B. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *In Proceedings of DARPA Image Understanding Workrhop*.

Luke, R., Keller, J., Skubic, M., and Senger, S. (2005). Acquiring and maintaining abstract landmark chunks for cognitive robot navigation. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2566–2571. IEEE.

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.

Meer, P., Mintz, D., Rosenfeld, A., and Kim, D. (1991). Robust regression methods for computer vision: A review. *International journal of computer vision*, 6(1):59–70.

Menegatti, E., Maeda, T., and Ishiguro, H. (2004). Image-based memory for robot navigation using properties of omnidirectional images. *Robotics and Autonomous Systems*, 47(4):251–267.

Milford, M. and Wyeth, G. (2008). Mapping a suburb with a single camera using a biologically inspired slam system. *IEEE Transactions on Robotics*, 24(5):1038–1053.

Milford, M., Wyeth, G., and Prasser, D. (2004). Ratslam: a hippocampal model for simultaneous localization and mapping. In *2004 IEEE International Conference on Robotics and Automation*, volume 24, pages 403–408. IEEE.

Milford, M., Wyeth, G., and Prasser, D. (2006). Ratslam on the edge: revealing a coherent representation from an overloaded rat brain. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4060–4065. IEEE.

Moller, R. (1999). Visual homing in analog hardware. *International Journal of Neural Systems*, 9(5):383–390.

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*.

Möller, R., Krzykawski, M., and Gerstmayr, L. (2010). Three 2d-warping schemes for visual robot navigation. *Autonomous Robots*, 29(3):253–291.

Möller, R., Lambrinos, D., Roggendorf, T., Pfeifer, R., and Wehner, R. (2001). Insect strategies of visual homing in mobile robots. In *In Proceedings of Computer Vision and Mobile Robotics Workshop*, volume 98, pages 37–45. Citeseer.

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., Gerstmayr, L., Röben, F., and Kreft, S. (2008). Neuroethological concepts at work: Insect-inspired methods for visual robot navigation. *Biological Approaches for Engineering*, pages 91–94.

Möller, R., Vardy, A., Kreft, S., and Ruwisch, S. (2007). Visual homing in environments with anisotropic landmark distribution. *Autonomous Robots*, 23(3):231–245.

Montiel, J. and Davison, A. (2006). A visual compass based on slam. In *2006 IEEE International Conference on Robotics and Automation*, pages 1917–1922. IEEE.

Müller, M. and Wehner, R. (2010). Path integration provides a scaffold for landmark learning in desert ants. *Current Biology*, 20(15):1368–1371.

Nagel, H. (1982). On change detection and displacement vector estimation in image sequences. *Pattern Recognition Letters*, 1(1):55–59.

Nordström, K., Barnett, P., and O'Carroll, D. (2006). Insect detection of small targets moving in visual clutter. *PLoS Biology*, 4(3):e54.

Papenberg, N., Bruhn, A., Brox, T., Didas, S., and Weickert, J. (2006). Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 67(2):141–158.

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.

Portelli, G., Ruffier, F., and Franceschini, N. (2010). Honeybees change their height to restore their optic flow. *Journal of Comparative Physiology A*, 196(4):307–313.

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.

Reinagel, P. (2001). How do visual neurons respond in the real world? *Current Opinion in Neurobiology*, 11(4):437–442.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Shi, J. and Tomasi, C. (1994). Good features to track. In *1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593–600. IEEE.

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., Lehrer, M., Kirchner, W., and Zhang, S. (1991). Range perception through apparent image speed in freely flying honeybees. *Visual Neuroscience*, 6(05):519–535.

Srinivasan, M., Zhang, S., Lehrer, M., and Collett, T. (1996). Honeybee navigation en route to the goal: visual flight control and odometry. *Journal of Experimental Biology*, 199(1):237.

Sturm, J. and Visser, A. (2009). An appearance-based visual compass for mobile robots. *Robotics and Autonomous Systems*, 57(5):536–545.

Stürzl, W. and Mallot, H. (2002). Vision-based homing with a panoramic stereo sensor. In *Biologically Motivated Computer Vision*, pages 3–14. Springer.

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.

Sunderhauf, N. and Protzel, P. (2010). Beyond ratslam: Improvements to a biologically inspired slam system. In *2010 IEEE Conference on Emerging Technologies and Factory Automation*, pages 1–8. IEEE.

Thrun, S., Burgard, W., and Fox, D. (2005a). *Probabilistic robotics*. Cambridge, MA: MIT Press.

Thrun, S., Burgard, W., Fox, D., et al. (2005b). *Probabilistic robotics*, volume 1. MIT press Cambridge, MA.

Vachhani, L. and Sabnis, A. (2011). Simple and robust correlation based visual compass using omnidirectional view. In *2011 IEEE Conference on TENCON 2011*, pages 1180–1184. IEEE.

Vardy, A. (2006). Long-range visual homing. In *2006 IEEE International Conference on Robotics and Biomimetics*, pages 220–226. IEEE.

Vardy, A. and Möller, R. (2005). Biologically plausible visual homing methods based on optical flow techniques. *Connection Science*, 17(1-2):47–89.

Vardy, A. and Oppacher, F. (2003). Low-level visual homing. *Advances in Artificial Life*, pages 875–884.

Vardy, A. and Oppacher, F. (2004). Anatomy and physiology of an artificial vision matrix. *Biologically Inspired Approaches to Advanced Information Technology*, pages 290–305.

Vardy, A. and Oppacher, F. (2005). A scale invariant local image descriptor for visual homing. *Biomimetic Neural Learning for Intelligent Robots*, pages 362–381.

Wehner, R. (1989). The hymenopteran skylight compass: matched filtering and parallel coding. *Journal of Experimental Biology*, 146(1):63–85.

Wehner, R. (1997). The ant's celestial compass system: spectral and polarization channels. *Experientia-Supplements Only*, 84:145–186.

Wehner, R. (2003). Desert ant navigation: how miniature brains solve complex tasks. *Journal of Comparative Physiology A*, 189(8):579–588.

Wehner, R., Michel, B., and Antonsen, P. (1996). Visual navigation in insects: coupling of egocentric and geocentric information. *Journal of Experimental Biology*, 199(1):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.

Yu, S. and Kim, D. (2012). Burrow-centric distance-estimation methods inspired by surveillance behavior of fiddler crabs. *Adaptive Behavior*, 20(4):273–286.

Yu, S.-E. and Kim, D. (2011a). Image-based homing navigation with landmark arrangement matching. *Information Sciences*.

Yu, S.-E. and Kim, D. (2011b). Landmark vectors with quantized distance information for homing navigation. *Adaptive Behavior*, 19(2):121.

Zeil, J. (1993). Orientation flights of solitary wasps (cerceris; sphecidae; hymenoptera). *Journal of Comparative Physiology A*, 172(2):207–222.

Zeil, J., Boeddeker, N., and Stürzl, W. (2009). Visual homing in insects and robots. *Flying insects and robots*, pages 87–100.

Zeil, J., Hofmann, M., and Chahl, J. (2003). Catchment areas of panoramic snapshots in outdoor scenes. *JOSA A*, 20(3):450–469.