Selection of Landmarks for Visual Landmark Navigation

Gert Kootstra[1]

stud.nr.: 1002848

September 2002

Supervised by:

Prof. dr. Lambert Schomaker[1]

Fumiya Iida[2]

[1] Artificial Intelligence, University of Groningen, Grote Kruisstr. 2/1, 9712 TS, Groningen, the netherlands. Email: [gert,lambert]@ai.rug.nl
[2] AILab, Department of Information Technology, University of Zürich, Winterthurerstr. 190, 8057 Zürich, Switserland. Email: iida@ifi.unizh.ch

Cover photo by: H.L. Leertouwer. Email: leertouwer@phys.rug.nl

Abstract

Insects are remarkably apt in navigating through a complex environment. Honeybees for instance are able to return to a location hundreds of meters away. What is so remarkable is that insects have this ability, although they have a tiny brain. This means that nature has found an accurate and economical way to deal with the navigation problem. The autonomous robots we use today face the same problem as the insects: These systems also need a good navigation ability, despite the fact that their computational power is limited. Robotics can learn a lot from nature. This is the field of biorobotics.

In the navigation task of finding back a location, bees use visual landmarks in the surroundings which pinpoint the location. But bees do not just use all the landmarks that are available in the surroundings of the goal location. They use the landmarks close to the goal for detailed navigation, since these landmarks best pinpoint the location. In order to select these nearby landmarks, a bee performs a turn-back-and-look behaviour (TBL). The image motion generated by the TBL provides the bee with information about the three-dimensional structure of the goal's surroundings. This information enables the bee to select reliable landmarks that are close to the goal location. When selecting the landmark, the bee learns the color, shape and size of the landmark, in order to be able to find back the goal location from any given other location from where the landmarks are visible.

We modeled this behaviour of using image flow to learn the useful landmarks in the goal's surroundings. To detect the motion flow we used an adapted version of the Elementary Motion Detector (EMD). The model is implemented on a freely flying robot, equipped with a omni-directional camera. The robot selects the reliable landmarks based on the image flow that appears when the robot is in egomotion.

Acknowledgement

In order to finish my Master's degree in Artificial Intelligence at the University of Groningen, I had to a graduation research. I have always had the desire to study abroad for a while and this was the perfect opportunity to fulfil this desire. Since I am interested in biologically inspired robotics, I soon found the AIlab of Rolf Pfeifer at the University of Zürich as the perfect place to the research. This master thesis is a result of the research that I have been working on in Zürich, from January 2002 until May 2002. After that period I returned to Groningen, to write the thesis. I would like to thank the following people for helping me with my graduation research:

prof. dr Lambert Schomaker, head of the Artificial Intelligence department at the University of Groningen, the Netherlands for the supervision of this graduation research.

Fumiya Iida, Ph.D. student at the AILab Zürich, Switzerland for the supervision of my graduation research during my apprenticeship in Zürich. And for the fact that thanks to his great dedication for his own research he interested and inspired me for scientific research.

Dr. Miriam Lehrer of the department of neurobiology at the University of Zürich, because she provided me with useful information about the visual navigation of bees and with information about the learning phase and the TBL behaviour.

Rick van de Zedde, a graduation student at the University of Groningen and a friend, with whom I had useful discussions about both our graduation researches.

All the members of the AIlab Zürich for the new insights they gave me in the field of Artificial Intelligence, from A-Life to passive-dynamic walking. And for a great time in Zürich.

Addresses

prof. dr. Lambert Schomaker
Director Research and Education
Artificial Intelligence
University of Groningen
Grote Kruisstraat 2/1
9712 TS Groningen
The Netherlands
Tel: +31-50-363-7908
Fax: +31-50-363-6784
E-mail: schomaker@ai.rug.nl

Dr. Miriam Lehrer
Dept of Neurobiology
University of Zurich
Winterthurerstr. 190
CH-8057 Zurich
Switzerland
Tel. +41 1 63549 75
Fax: +41 1 635 57 16
Email: miriam@zool.unizh.ch

Fumiya Iida
AIlab, Department of Information Technology
University of Zurich
Winterthurerstr 190
CH-8057 Zurich
Switzerland
Tel: +41-1-635-4343
Fax: +41-1-635-6809
E-mail: iida@ifi.unizh.ch

Table of Contents

Chapter 1

Introduction

This is a study in the field of *biorobotics.* Biorobotics is the field where biology meets robotics. On the one hand, biologists can use robotics for verification, by testing their theories on robots. On the other hand, and that is important for us, roboticists can use the results from biology studies for the control of their robots. Nature has provided many excellent examples of autonomous agents through millions of years of evolution. Since in robotics we try to construct artificial autonomous agents, we can learn a lot from nature.

In this research project, we look at the navigation strategies of insects, to gain inspiration for the navigation of a *flying robot.* An excellent navigation capability is essential for insects in order to survive. Therefore, nature provided them with very good strategies for this purpose. Studying biological navigation strategies has a number of advantages. Insects have a very small brain. The brain of a bee is not bigger than 1 mm^3 and contains about 10^6 neurons, whereas the human brain consists of 10^{11} neurons. This means that insects are not capable of performing complex calculations. Despite this limitation, insects are still capable of performing excellent flight navigation. This means that the strategies are *computationally cheap.* Further more, insects navigate *completely autonomous*, this means that we can find a strategy that does not depend on external systems, like the GPS system. Thus, we could find and develop a strategy that works in a great number of environments, even in places where these external support systems are not available. Furthermore, the navigation strategies of insects hardly ever fail, in other words, they are highly *robust.* By studying the navigation strategies of insects, we would like to develop a navigation strategy for robots that is *computationally cheap*, *autonomous* and *robust.*

Bees have many strategies for navigation. They make use of proprioceptive information (for instance the number of wingbeats) for information about the distance they traveled. Bees use the polarization of sunlight on the sky for their orientation. The earth's magnetic field gives information about orientation, as well as information about the position of the bee, through the small changes in the magnetic field on different positions. However the most important strategies are based on *vision.* Vision is used to maintain a straight course, to control the speed, to provide a safe landing, to avoid obstacles and for *landmark navigation.*

Bees use visual landmarks, salient objects in the surroundings, for localization. Based on the visual landmarks, the bees know where they are and how they should reach their goal. Much is known about landmark navigation (e.g. [Cartwright and Collett, 1983; Wehner, Michel, and Antonsen, 1996]) and many implemented methods for navigation are based on this principle (for instance [Franz et al., 1998; Lambrinos et al., 2000]), but, although it is clear that not all objects in the environment are selected as landmarks, most studies do not discuss the learning phase of the landmark navigation. We wonder: *Which objects are chosen as landmarks during the learning phase* and *how are these landmarks selected?*

In this present study, we want to answer these two questions to have an understanding in *how we could implement the learning phase of the landmark navigation on an autonomous flying robot in a computationally cheap and robust way.* A good understanding of this problem would give a flying robot the ability to learn a location in an unknown environment, so that, the robot is able to return to that location. To find the solutions to these questions, we will first have a look at the biological backgrounds of this problem. Thereupon we will propose a model for the implementation on the autonomous flying robot. Next we will discuss the practicability of these model by means of some experiments and the results of these experiments. Finally we will discuss the advantages and disadvantages of our model as well as the possibilities for future work.

Chapter 2

Insect Navigation

In the present study we try to learn from nature when designing robotic systems. This process is called: From animals to *animats*. An animat is a simulated animal or autonomous robot. In this chapter we will look at navigation strategies found in biology studies. The ability to navigate in the complex world is probably the most basic requisite for an animal's (and an animat's) survival. Without that ability, the animal (or animat) would not be able to reach food (energy) sources, to avoid damaging obstacles, or to escape from dangerous predators.

We will discuss insect navigation based on vision, in particular *landmark navigation*. This is the behaviour of insects to find back a location, which they visited before, based on visual salient objects in the environment, so called *landmarks*. The main subject of our study is which landmarks are used to navigate on and how these landmarks are selected.

2.1 Visual Navigation and Movement Control

Insects use different kinds of navigation strategies. Ants, for instance, make use of *trace-based navigation*. They make pheromonal trails on their way which they can smell and use to find back the nest or a food source. Another strategy is *Dead reckoning*. With this method the position in the world is constantly updated by summing successive small displacements with respect to the body orientation, which is called *path integration*. The insects know the orientation of their body by sensing the earth-magnetic field [Gould, 1980] or by using polarization patterns in the sky to gain compass information [Wehner, 1982]. The displacement of their body can be estimated by using proprioceptive information, like energy consumption or some kind of 'step counting' [Ronacher et al., 2000]. A third navigation strategy is *gradient-based navigation*: The gradient of the sensory input is used to navigate through the environment. Bees for instance are sensitive to the small fluctuations in the magnetic field, which form an unique pattern for every location [Gould, 1980]. Many fly subspecies use the gradient in temperature to navigate to warm places, in order to find a warm body from which they can suck blood.

But most insects use visual input as the main source for navigation and movement control. Especially aerial insects, like flies and bees, heavily rely on vision during the navigation, considering that they can not make use of pheromonal trails and the proprioceptive errors are much larger in the air then on land. Past research shows that flying insects use vision to control their flight. They use vision to maintain a straight course during their flight [Reichardt, 1969], to control the altitude [Mura and Franceschini, 1994], to regulate the speed of their flight [Srinivasan et al., 1996], to achieve a smooth landing [Srinivasan et al., 2000], for obstacle avoidance [Srinivasan et al., 1996] and for odometry (to measure how far one has traveled) [Esch and Burns, 1996; Srinivasan, Zhang, and Bidwell, 1997; Srinivasan et al., 2000]. These movement control strategies are all based on image flow. The details will be revealed further in section 2.1.3.

Beacon navigation is another visual navigation strategy, where the insect locates a beacon and directly navigates towards this object. But in this research, we are interested in the strategy that is called *landmark navigation*. In this strategy the goal is to find back a home location that is not directly visible itself. Landmarks in the environment that surround the home location are used to the navigate towards the goal. This strategy will be further discussed in section 2.1.2. But first we will discuss how insects receive visual input. In particular we will have a look at the bee's compound eye.

2.1.1 The Compound Eye

The bee's eye, like the eye of many insects, is built quite differently from the vertebrate eye. The eye of vertebrates consists of a single lens, which focuses the image on a light-sensitive 'film', the *retina*. The bee, on the other hand, has an eye consisting of a great number of *facets*, called a *compound eye*. Worker bees have about 4500 facets in each eye. Each facet is an independent eye aimed at an unique part of the visual world. Below each facet is an individual light gathering structure, called an *ommatidium*, which records a general impression of the color and intensity of the light from the direction in which the facet faces. The *retinula cells* at the bottom of the base of each ommatidium are the sensors that convert the different properties of the light to an electrical impulse, stimulating the bee's brain. All the impulses from the individual ommatidia are pieced together for the overall picture.

The resolution of the bee is quite poor. For a comparison, the bee's brain receives one percent as many connections as the human eye provide. Although the compound eye cannot register fine detail, it is excellent at detecting motion: The image processing is so much more efficient than is the case with the human eye, that the compound eye offers a much greater *flickering fusion rate*. Bees can notice flicker up to 200 Hz, whereas human can only see up to 20 Hz. This means that the bee can detect slight changes in his visual field much more quickly than human.

Another remarkable property of the eyes of the bee is that they are placed on the sides of his head, which allows the bee to look all around. The bee has nearly a 360^o visual field in the horizontal plane. Consequently, a large part of the visual field is only covered by one eye, especially at the *lateral part* (the left and right side of the bee). This means that the bee cannot rely on stereo vision to gain a three dimensional (3-D) perspective, like we human do. Our brain is able to combine the two slightly different views from each eye to produce 3-D perception. Even in the parts where there is overlap of the visual fields of the bee's eyes, 3-D perception is not reliable enough, because the distance between the eyes is too small to sense a difference in the view.

Besides the two compound eyes, the bee possesses three other photoreceptors, the *ocelli*, located on top of the bee's head. These receptors are sensitive to polarized light. The sunlight gives a polarization pattern in the sky, by detecting this pattern, the bee knows his

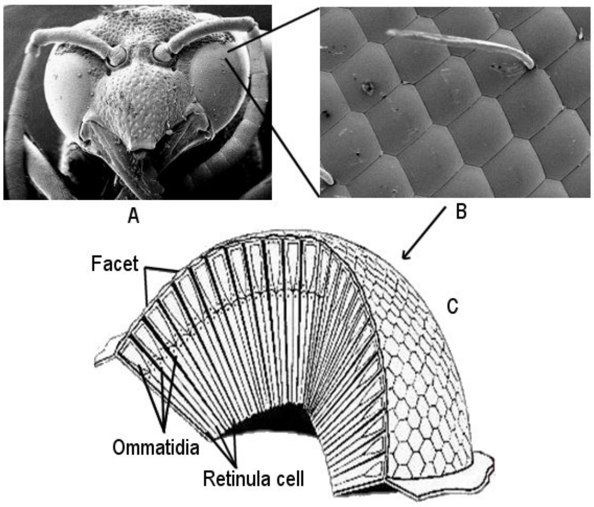

Figure 2.1: *The bee's compound eye. (**A**) shows a front view of the bee's head, magnified 35 times. A part of the eyes are shown in (**B**), 1000 times magnified. (**C**) shows a section of the eye, where each individual facet is shown, with the ommatidium collecting the light and the retinula cell converting the different properties of the light to electrical impulses.*

orientation.

Pathways for Motion Detection

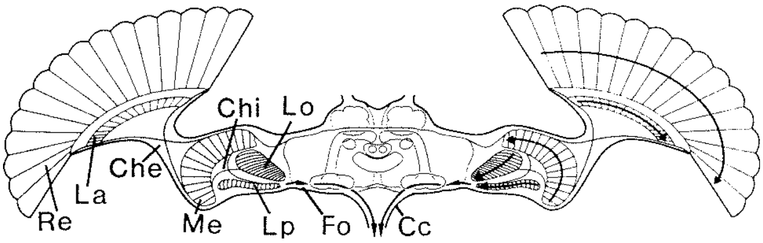

Figure 2.2: *The visual system and brain of the fly. The optic lobes subserving the retina (Re) of each eye consist of the neuropils lamina (La), medulla (Me), lobula (Lo), and the lobula plate (Lp), which are connected by the external and internal chiasm (Che, Chi). The visual neuropils show retinotopic columnar organization as indicated by the arrows in the right eye and optic lobe. Outputs of the lobula plate project into the optic foci (Fo). The optic foci is connected with the motor centers in the insect's brain through the cervical connective (Cc). Figure from [Hausen, 1993].*

The detection of motion in the visual field is an important property of the insect's eyes and the underlying optic lobes. Figure 2.2 shows a section of the eyes and the optic lobes. Here we will describe the neuronal pathways for detecting motion in the visual field.

The retina consists of the separated facets, each with his own retinula cells (photoreceptor). The organization of the facets remains throughout the processing of the input signals in the *lamina*, *medulla* and *lobula* neuropils. Amacrine cells are post-synaptic to the retinula cells. Detecting motion requires lateral connections, connections between adjacent facets. This is provided by T1-, L2- and L4-cells, whose dendrites get input from adjacent amacrine cells. The amacrine, T1, L2 and L4 cells are located in the *lamina.* The transmedullary cells, Tm1, is located in the *medulla.* This cell receives input from the T1-, L2- and L4-cells. Tm1 is sensitive to motion in a preferred direction. It is insensitive to motion in the opposite direction. The axons of Tm1 cells terminate onto T5 cells in the *lobula.* The T5 cells receive input from different motion detection cells (i.e., the Tm1 cells), with opposite preferred directions. The T5 cells are motion detectors, which are sensitive to motion in both directions. See [Douglass and Strausfeld, 2001] for more detailed information.

The above described structures accomplish local elementary motion detection. In 1969, Reichardt proposed a model for motion detection based on behavioral studies on insects, the Elementary Motion Detector (EMD) model [Reichardt, 1969]. Years later, microscopic studies on the visual system and brain of insects show that the structure and functionality of the EMD model strongly resembles that of the natural system which we described above. We will describe the EMD model in detail in section 3.2.

In the *lobula plate*, the motion detection signals of the T5 cells are further processed. Wide field neurons, so called tangential cells, receive input from many T5 cells. There are two classes of giant tangential cells. The horizontal system (HS) consists of three cells, having dendritic input from the dorsal, medial and ventral part of the visual field. The vertical system consists of eleven cells with vertical dendritic fields, which together cover the entire visual field. These tangential cells terminate in the optic foci, where the information about global as well as local motion is passed on to pre-motor neurons. See for more information [Hausen, 1993].

2.1.2 Landmark Navigation

It is shown that during the bee's navigation task of finding back a home location (so called *visual homing*), it is guided by salient objects in the environment. These salient objects are called *landmarks.* On of the best known researches on landmark navigation is done by Cartwright and Collett [Cartwright and Collett, 1983]. The purpose of their study was to have a better understanding about how bees use landmarks to find the goal location.
Bees were trained to collect sugar water in a completely white room. The hive of the bees was outside the room, they had to enter the room through a window. The sugar water itself was not visible, but was marked by one or more landmarks at a certain orientation and distance from the food source. After each of the bee's foraging trips to the room, the landmarks and the food source were moved as a group to another part of the floor, keeping the same orientation and distance from the landmarks towards the food source. This was to prevent the bees from expecting the food source in any particular area.

After half a day of training, test were given. The bees arrived to find the array of landmarks present, but the food source missing. The bees than started a search flight. This flight was recorded and every 100 ms the location of the bee was marked. The location with the highest search density was said to be the location where the bee expected the food source.

At one experiment, the bees were learned to associate the food source with a single landmark. During the tests, where the same landmark was placed at different locations in the room, the bees searched exactly at the location where normally the food source would be, with the same orientation and distance towards the landmark. If the bees had no sense of direction, the highest search density would be equally divided on a circle around the landmark. Since this was not the case, it can be concluded that the bees use compass information (e.g., the earth's magnetic field or the polarization patterns on the sky) for their orientation. The bees searched at the right distance from the landmarks, apparently the bees had also learned the distance of the landmarks from the location of the food source. Bees cannot use stereo vision for distance information, what remains are two possible strategies for gaining the distance towards objects: In the first place, the apparent size of the objects can be used (i.e., the size of the object as it appears at the retina[1]). The closer an object is to the bee, the bigger the object appears. Secondly, the distance information can be gained by using the angular velocity at which objects moves across the retina when the bee flies by. The closer an object is to the bee, the faster the object will move across the bee's retina. To test this, a second experiment was set up.

Again the bee was trained to collect sugar water from a location marked by a single landmark. This time, during the tests, a bigger landmark was placed in the room. Now the bees searched in the right orientation, but at a distance father away from the landmark. Exactly at the distance where the landmark appeared with the same size on the bees' retina as during training. This clearly shows that the bees use the apparent size of the object to gain distance information.

From the results of this study, Cartwright and Collett proposed the *snapshot model*, a model that widely agreed on and used in many studies (e.g., [Moller et al., 2000; Franz et al., 1998; Trullier et al., 1997; Bianco, 1998]).

The Snapshot Model

When a bee is at a location that it wants to revisit, the goal location, the bee takes a *snapshot*. This means that the bee stores information about the size and position of the landmarks in the surroundings. The angular position of the landmarks on the retina is stored, as well as the apparent size of the landmarks, the angular size that the landmarks have on the retina. Apart from the information about the landmarks, the angular position and apparent size of the gaps between the landmarks is stored as well.

When the bee is displaced from the goal location, the position and size of the landmarks change. This can be seen in Figure 2.3, where the black areas in the inner circle give the position and size of the landmarks as taken in the snapshot and the grey areas in the second circle give the position and size of the landmarks as they currently appear on the retina. The white areas correspond to the gaps between the landmarks.

A home vector. pointing approximately to the target position, can be derived from pairing each area in the current view with the closest sector of the same type (landmark or gap) in the snapshot, where snapshot and current view are aligned in the same compass direction. Each pairing generates two vectors. A tangential vector pointing so as to align the position of the two areas, from the area in the snapshot to the corresponding area in the current view, because the agent needs to turn in that direction to align the positions. And a radial vector, pointing so as to match the size of the corresponding areas. Pointing outside if the size in the current view is smaller than in the snapshot, because the agent needs to come closer to

[1] Although the bee does not have an eye with a single retina, but an compound eye consisting of many 'retinae', we will talk about 'the retina' in the proceeding chapters of this thesis, for simplicity.

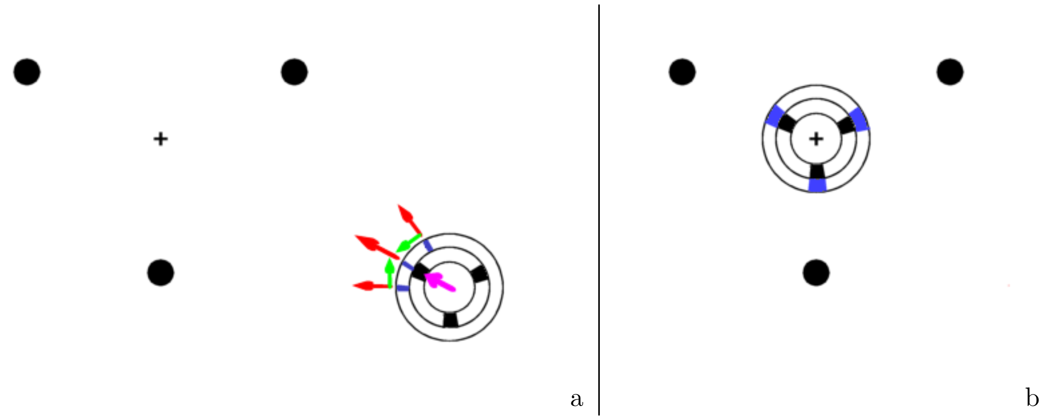

Figure 2.3: *The snapshot model. This is a top view. The bee is represented by the three circles, since a bee can see all around. The bee has stored the position and size of the landmarks,* ●*, on the retina as seen from the home location, +. This is called a* snapshot*. The snapshot is represented by the black areas in the inner circle. The blue areas in the second circle indicate the position and size of the landmarks as they currently appear on the bee's retina. (**a**), the bee is at a distance from the home location. Each landmark and gap between the landmarks in the current view is now compared with the best match in the snapshot. A tangential vector (green) is created to align the positions (e.g. if a landmark should be more to the right, the bee has to turn left). A radial vector (red) tries to match the sizes between current view and snapshot. Pointing outside is the size in the current view is too small and vice versa. The home vector (purple) is the result of the summation of all the individual vectors and points towards the home location. (shown smaller in the figure). (**b**) shows the bee at the home location. The current view and snapshot are aligned.*

match the sizes, and pointing inside when the size in the current view is bigger. The home vector is derived by summing all the individual vectors.

Möller argued that bees would not have enough memory capacity to store a complete snapshot. He therefore proposed the *Average Landmark Vector* (ALV) model, which is based on the snapshot model, but more economical. See [Moller, 2000] for more detail on the ALV model.

2.1.3 Movement Control by Using Image Flow

An important group of movement control strategies of insects is based on *image flow*. With image flow we mean the motion of objects on the retina (or in the camera image of animats). When an insect moves (i.e., when it produces ego-motion) all the objects in the surroundings move on the retina. This image flow can tell the insect much about the layout of the environment and about his own flight, about the speed, the altitude and the rotation of the flight. To explain the variety in the use of image flow in insects navigation, we will discuss a few studies on this subject.

Obstacle Avoidance and the Centering Response

Bees, like most insects, possess very small inter-ocular separations and therefore cannot rely on stereoscopic vision to measure distance to objects or surfaces. Despite this fact, the bee is capable of flying through the middle of a gap, which can be seen as obstacle avoidance, with

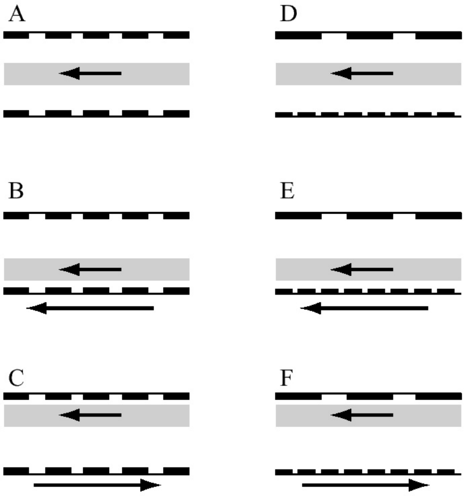

Figure 2.4: *Illustration of an experiment demonstrating that flying bees infer range from apparent image speed. The short arrows depict the direction of the flight and the long arrows the direction of grating motion. The shaded areas represent the means and standard deviations of the positions of the flight trajectories, analysed from video recordings of several hundred flights. From [Srinivasan et al., 1996].*

the walls as obstacles. Srinivasan set up a research to see how bees solve this task [Srinivasan et al., 1996]. Figure 2.4 shows the experimental setup of this study. A sugar solution was places at the end of a tunnel. The bees were trained to collect the sugar by flying through the tunnel. Each side of the wall carried a pattern consisting of a vertical black-and-white grating. The grating on one wall could be moved horizontally to both sides.

When both gratings were kept stationary, the bees flew through the center of the tunnel, i.e., they maintained equidistance to both walls (Fig. 2.4 A). But when one of the gratings was moved at a constant speed in the direction of the bees flight (thereby reducing the speed of image flow on the eye facing that grating) the bee's trajectories were shifted to the side of the moving grating (Fig. 2.4 B). When the grating was moved in the opposite direction (thus increasing the speed of image flow), the trajectories were shifted away from the moving grating (Fig. 2.4 C). This suggests that the bees keep equidistance to both wall by balancing the apparent angular speeds of both walls, this is balancing the speed of image flow in both eyes. A lower image speed on one eye was evidently taken to mean that the grating on that side was farther away and causes the bees to fly closer to that side. Srinivasan even could make the bees bump into the wall.

To be sure that the bees balanced the speeds on both sides and not the black-and-white frequency of the gratings at both sides. Gratings with different spatial periods were placed on the walls (Fig. 2.4 D,E,F). This did not influence the bees' flight trajectories, thereby proving that the bees really used the speed of image flow on left and right eye to keep the walls at equidistance.

Regulating Speed

In a similar way Srinivasan et al. [1996] showed that bees regulate the speed of their flight by monitoring the apparent velocity of the surrounding environment. Just as in the previous study, the bees were trained to fly through a tunnel, but this time the tunnel was tapered. The flight speed of the bees was exactly so that the apparent speed of the gratings on the walls maintained constant. The bees slowed down when approaching the narrowest section of the tunnel and accelerated when the tunnel widens beyond it. The advantage of this strategy is that the bees automatically slow down when approaching a difficult narrow passage.

Grazing Landing

With exactly the same strategy bees also perform a smooth landing on a horizontal surface [Srinivasan et al., 2000]. When the bee approaches the surface, the apparent speed of the texture on the surface increases. The bee tries to keep a constant apparent speed and slows down. Therefore the speed of the bee is close to zero at touchdown. The advantage of this strategy is that control of flight speed is achieved without explicit knowledge about the height.

Odometry

For a long time the thought was that bees use the amount of energy consumption or the number of wingbeats to measure the distance flown (i.e., proprioceptive information). But since bees and other arial insects fly and are subject to unknown winds, this strategy seemed not to be a reliable measurement of distance flown for these animals. A headwind, for example, would then give a bee the impression that it has covered more distance. But how do bees measure distance flown? The experiments in [Srinivasan, Zhang, and Bidwell, 1997; Srinivasan et al., 2000] showed that bees gain distance traveled by integrating the apparent speed of the surroundings.

Bees were trained to collect food at the end of a tunnel, which had vertical black-and-white patterns at the walls. After a few hours of training, the food source was removed, to test the bee. The next time the bee was to collect the food, it searched for the food in the tunnel. The location with the highest search density was considered to be the place where the bee expected the food source. When the period of the strips in the tunnel was double or half the period during training, the bee expected the food at the correct distance from the tunnel entrance. So the bee did not estimate the distance by counting the number of strips. However, when the bee was tested in a wider tunnel, the bee searched at a greater distance, while it searched closer to the entrance when the tunnel was narrower. The places where the bee expected the food source matched the place that one would expect when the bee measured distance flown by integrating the apparent speed of the black-and-white gratings. Because flying through the wider tunnel produced less speed of the stripes, the bee flew further in order to compensate for this.

These different studies demonstrate the importance of image flow for insect navigation. Insects use the image flow in different parts of the visual field for different purposes: The lateral parts are used for the centering response, for odometry and for regulating the speed. For a smooth landing the caudal (under) part is used. This part is also used to regulate of the speed. For avoiding obstacles which the insect is approaching, the image flow in the frontal part is used.

2.2 Landmark Learning in Bees

When bees, like most insects, want to be able to return to a certain location, they learn the environment by the landmarks in the environment. On the point of returning, the landmarks can pinpoint the goal. However, the bees can use many landmarks. The question is if the bees use all available landmarks for the navigation. If not, which landmarks do bees use and how do they select these landmarks?

2.2.1 Which Landmarks are Used?

In [Cheng et al., 1987] bees were trained to collect sucrose from a place surrounded by an array of highly visible cylindrical landmarks. After a few hours of training, the bees were tested singly on a test field containing the array of landmarks (with some modifications), but in the absence of the sucrose. In that case, the bee searched at the position where it expected the sucrose. The position of the bee was recorded four times per second.

When the bees were trained with two landmarks close to the goal location and two landmarks further from it, and were tested with only two landmarks, they searched for the food source at the position where it should be relative to the two nearby landmarks (see Figure 2.5a). They were not guided by the landmarks further away from the goal. There are two possible

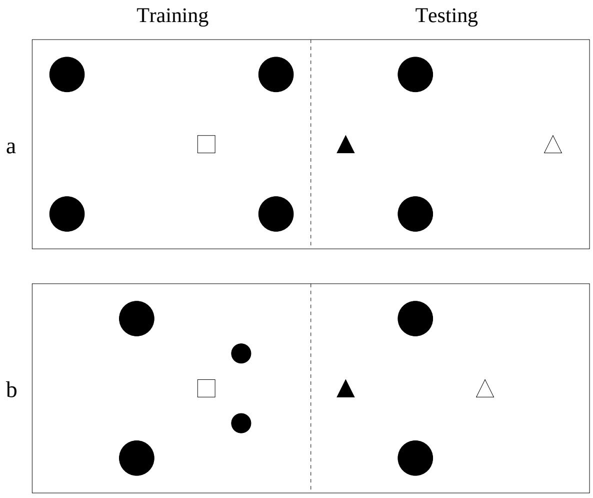

Figure 2.5: *(***a***) The bees were trained for a few hours to collect sucrose , □, from a location marked by two near and two distant landmarks, marked by ●, both of the same size. In the test, only two landmarks are placed in the room. The highest search density was at ▲. This is the location where one expects the bees to search for the sucrose if they were guided by the landmarks closest to the goal during training. If the bee would be guided by the two distant landmarks, the highest search density would be at △. (***b***), the bees were trained with two small landmarks close to the goal and two large ones further from the goal. The apparent size of all four landmarks was equal at the goal location, □. The bees were tested with two large landmarks placed in the room. If the bees would us the landmarks with the largest apparent size (all four) they would search both at the left and right side of the landmarks. But the test showed that the highest search density was at ▲ and not at △. This clearly shows that the bees search at the position as specified by the landmarks closest to the goal during the training.*

reasons why the nearby landmarks are used to pinpoint the goal. Was that because of the distance of the landmarks from the goal or because of the apparent size of the landmarks as viewed from the goal location (the nearby landmarks appear bigger on the bees' retina)? To explore this question, the bees were trained with two small landmarks placed near the goal and two bigger landmarks further away from the goal, in such a way that the apparent size of all the landmarks was equal at the goal location. During the tests the bees were again guided by the nearby landmarks. The bees did memorize the landmarks further from the goal, but the nearby landmarks were highly preferable (Figure 2.5b).

In [Collett and Zeil, 1997], it is also concluded that bees are sensitive to the absolute distance towards objects and that they prefer to be guided by objects near to the goal for detailed navigation. But they added that bees also use distant landmarks for long distance navigation. These landmarks have to be large in order for the bee to see them from a great distance. Distant landmarks are visible in a large area, but because of the great distance of the landmarks and the limited resolution of the bee's eye, these landmarks can not provide in detailed navigation, but they can only guide the bee near the goal area. Landmarks close to the goal are only visible in a small area and therefore can not be used for navigation over long distances, but they can much more precisely pinpoint the goal location.

2.2.2 How are the Landmarks Selected?

It is evident that bees are sensitive to the distance of objects, but the question remains: How do bees measure the distance towards objects in order to select them for the landmark navigation task? As mentioned in section 2.1.2, there are two possible answers to this question: The distance can be gained by using the *apparent size* of the objects, as was the result in the snapshot model [Cartwright and Collett, 1983]. The bee can only use this strategy when it has a priori knowledge about the absolute size of the landmark, for 'bigger is closer' is not always true. For example a huge building far from the viewer can appear bigger than a nearby pencil.

The second possible answer is to use the *angular velocity* of an object when the bee is in ego-motion. Everybody who has been in a train noticed at least ones that the poles at the sides of the railway go by the window really quickly, whereas, for instance, a cow far away in a meadow passes much slower. The faster an object seems to move across the retina (i.e., the higher the angular velocity), the closer the object is. This strategy, that is based on the speed of the image motion, requires a *constant speed* of the observer during the selection phase, since the speed of the ego-motion also influences the angular velocity of objects. Further more this strategy requires a *stable course*. This will be explained in more detail in section 2.3.

Turn-Back and Look behaviour

Lehrer has been studying bees for many years. In one study she looked at the *learning phase* of landmark navigation. That is the first few times that the bee visits the home location (e.g. a flower). It is the phase in which the bee learns different cues about the surroundings at the home location. During one of her experiments she noticed a remarkable behaviour of the bees during the learning phase: The bees performed a so called *Turn-Back and Look* behaviour (TBL) [Lehrer, 1993].

When the bee departs from a flower during the learning phase, it does not fly to the hive in a straight line. The bee flies a short distance away from the flower, then turns around and looks back to the home location, while hovering sideways, to left and right. The bee then continues

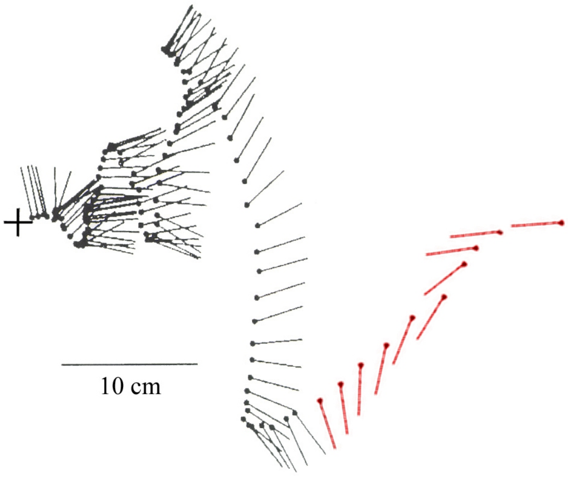

Figure 2.6: *TBL of a bee leaving a food source (+). The position of the bee's head (filled circle) and body orientation (line) are shown every 20 ms. The TBL behaviour is shown in black. After the TBL, the bee faces towards the hive and flies away in that direction (shown in red). Figure from [Lehrer and Collet, 1994]. The red part is added, to clarify the behaviour of the bee after the TBL.*

to fly away from the goal, still looking back to the flower. This behaviour is performed for a short time before the bee flies in a straight line to the hive (see Figure 2.6). The bee shows this behavior only when departing the goal and only in the first few flights from the flower to the hive. When the learning phase is over, this behaviour is not performed anymore.

Why do bees perform TBL behaviour? Probably they turn back and look while they depart the goal area, to look at the area and learn information about it, in order to be able to return. A research on what bees learn about the goal area during the TBL-phase [Lehrer, 1993] resulted in the conclusion that bees learn the *size*, *shape* and *color* of the objects surrounding the goal mainly when *approaching* the goal, so not during TBL. And that they observe the *distance* towards these objects only during the *TBL-phase*. This distance information is inferred from the *speed of the image flow* that is observed by the bees during the TBL behaviour.

Srinivasan et al. [1989] also concluded that honeybees measure the distance from an object of unknown size by using its apparent motion across the retina. The distance information can be used to segregate the view into those features that might be useful for navigation and those that are irrelevant. Lehrer showed that the color of an object close to the goal is learned better than that of a more distant one with the same apparent size [Lehrer, 1993]. Similarly, bees learn the size of an object better when the object is near than when it is further [Lehrer and Collet, 1994]. In other words, objects close to the goal are selected as landmarks. The features of these landmarks that are learned, are the direction and size viewed from the goal, the color and, less important, the shape. The reason that the nearby objects are selected is that they best define the goal location.

In one of the experiments, which clearly shows that during the TBL-phase bees use image flow to infer distance information, bees were trained to collect sugar water from a location marked by a single landmark. The experiment was identical to one of the experiments of

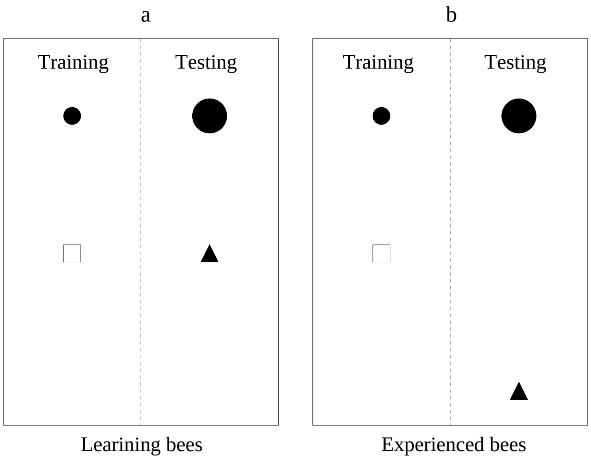

Figure 2.7: *In both experiments, the bees were trained to collect sugar water from a location,* □, *marked by a single landmark,* ●. *In experiment (***a***), the bees were trained only for a few trials, so that they were still in the learning phase. When the bees were tested in a test field with a bigger landmark and in the absence of the sugar water, they searched at the location with the same orientation and distance towards the landmark,* ▲. *At a distance where the landmark produced the same amount of image flow as during training. Experiment (***b***), is an identical experiment, but now the bees were trained for half a day and so were experienced. The bees now searched at a location with the same orientation, but with a larger distance towards the landmark,* ▲. *At a distance where the landmark appeared at the same size as during training.*

Cartwright and Collett, except that this time the bees were not trained for many hours, but just for a few flights. During the test, a bigger landmark was placed in the test field. This time the bees searched for the food source at the correct distance from the landmark, where the image flow of the landmark was equal to that during training, although the apparent size of the landmark was much bigger (see Figure 2.7).

So we see two distinct phases, the *learning phase* and the phase when the learning phase is over, the *landmark navigation phase.* During the learning phase, when the landmarks are selected and learned, bees use image flow to gain a 3-D perspective of the world by performing TBL. During the landmark navigation phase, when the sizes of the landmarks are already learned, the bees use the apparent size of the object to measure the distance. The reason for this shift in the bee's behaviour is that the use of image flow for the absolute distance is more cumbersome than relying on the apparent size, because on each approach the returning bee would need to scan the scene as it did during the TBL phase.

2.2.3 Which Landmark Cues are Learnt?

Which information about the landmarks does the bee learn? From [Cartwright and Collett, 1983] we know that the bee remember the apparent *size* and the *position* on the retina as view from the home location. By using the apparent motion of objects the bee gains information about the *distance* [Lehrer, 1993; Srinivasan et al., 1989]. Lehrer tells us that the bee also remembers the *color* of the landmarks [Lehrer, 1993]. And at last we know that bees learn the *shape* of the landmarks [Lehrer, 1993; Hateren, Srinivasan, and Wait, 1990].

2.3 Course Stabilization

There are two requirements in order to be able to use image motion to gain distance information. The observer has to move with a constant speed, because the speed of ego-motion strongly influence the amount of image flow. It is observed that bees have more or less a constant speed during the TBL-phase [Lehrer, 1993], so that requirement is satisfied. The second requirement is that the observer has to move in a straight line, a translational movement without any rotations. Only with a translational movement the concept of 'when the image motion is faster, the object is closer' is valid. When the observer makes a purely rotational movement, all the objects in the environment, near or distant, move with the same speed across the retina, and so can not be used to gain distance information. A Movement that is both rotational and translational (e.g. a wide turn) does not provide an useful image flow pattern as well. A turn to the left, for instance, provides a higher speed of image flow at the right side than at the left, which make objects at the same distance look closer at the right side than at the left side. Do insects satisfy this second requirement as well?

2.3.1 A Detailed Analysis of the Insect's Flight

"Have you ever seen a fly circle around a lamp?". Most people will give a positive answer to this question. Schilstra [1999] performed an analysis on insects to get more insight in the details of the flight. He placed tiny sensors on the head and thorax of a housefly, which accurately measured the position and orientation of the head and thorax. With this information, the flight of the fly was registered. The results showed that a fly does not circle around a lamp, but that it 'squares' around a lamp.

The fly tries to fly in a straight line as long as possible. When he has to make a turn, the thorax starts the rotation, while the head compensates this, remaining the same orientation as before. After a while the head also starts to rotate in the same direction as the thorax, but much faster. The orientation of the head overtakes that of the thorax and after a while the head gets a fixed orientation, while the thorax is still rotating. Finally, when the orientation of the thorax is equal to that of the head, the turn is ended. Figure 2.8 shows the orientation of the head and thorax when the fly makes a turn. This observation clearly shows a strategy of the fly to keep the orientation of the head stable as long as possible, by reducing the period of rotation to the minimum. The fly flies more or less in squares.

So insects also satisfy the requirement of a stable course for using image motion to gain a 3-D perspective of the world. Schilstra also notes that the fly performs the described strategy to be able to get as much information about the environment by using image motion as possible.

2.3.2 How to Stabilize the Flight

To have a full understanding of the selection of landmarks strategy of insects, we need to know how insects stabilize their flight in order to gain correct distance information based on image flow. According to Reichardt and Poggio [1976] and Heisenberg and Wolf [1984] insects stabilize their flight again by using image motion. With the so called *optomotor response* they stabilize the course of their flight. When an insect is placed in a cylinder with black-and-white patterns on the wall and the cylinder is rotated, the insect tends to turn with the movement. When an insect flies not in a straight line, but makes a turn to the left, it notice this, because the angular velocity of the surroundings at the right side is higher

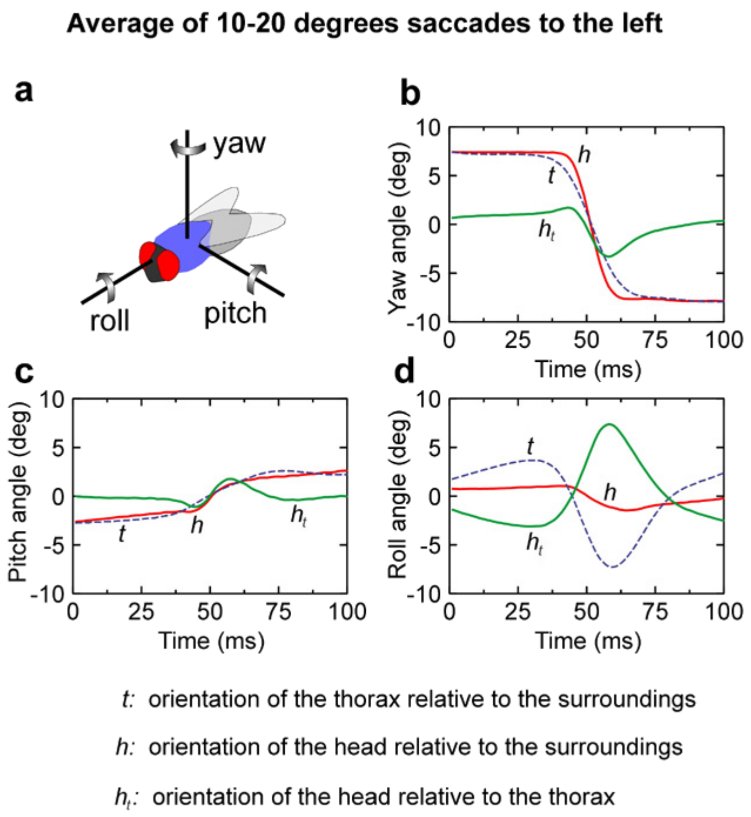

Figure 2.8: *The orientation of the thorax (blue dashed line, t) and the head (red line, h) during a turn. During the turn, the thorax makes a roll, the head almost completely compensates for this. The head also compensates thorax rotation around the vertical axis (yaw, both at the beginning and at the end of the turn, in the middle of the turn, the yaw of the head is faster than that of the thorax. The result is that the period of rotational movement is reduced, thereby increasing the period of translational movement. Thus increasing the period in which the insect gains depth information. Figure from [Schilstra, 1999].*

than that at the left side. Subsequently, the insect compensates for this by turning to the right in order to stabilize the flight.

2.4 Conclusion

We can conclude that bees use vision, especially image flow, for many navigation and movement control strategies. Landmark navigation is an important visual navigation strategy. Bees use landmarks to guide them to a goal. Bees do not use all available objects in the surroundings. During the learning phase, they select objects near the goal as landmarks. These nearby objects are selected by using image motion. The device is: The higher the angular velocity of an object, the closer the object is. The preconditions to obtain distance information from image flow are that the bees maintain a constant speed and a stable flight. Bees stabilize their flight again by using image flow.

In the next chapter, we will use the biological findings about the landmark selection to propose a model for landmark selection in an autonomous flying robot. The model consists of two submodels. The flight-stabilization model and the actual landmark-selection model. Both models are based on the biological studies that we outlined in section 2.2 and 2.3.

Chapter 3

A Visual Landmark-Selection Model for Flying Robots

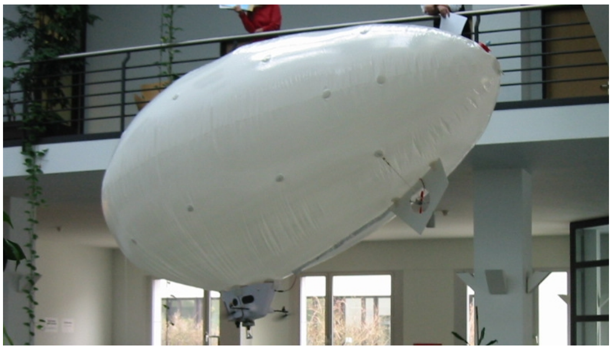

Figure 3.1: *Melissa, the flying robot platform*

3.1 Melissa

The models we will propose in this chapter, will all be implemented on a flying robot platform called *Melissa* (see Figure 3.1). Melissa is a blimp-like flying robot, consisting of a helium balloon, a gondola hosting the on-board electronics, and a off-board host computer. The balloon is 2.3m long and has a lift capacity of approximately 400g. The gondola hosts the electronics for the perception and action of the robot.

3.1.1 Perception

Insects use their vision to select landmarks. Therefore we equip Melissa with an omni-directional vision system, which provides the sensory input (see Figure 3.2).

The omni-directional vision system consists of a CCD-camera placed in front of a hyperbolic mirror, based on a panoramic optics studie [Chahl and Srinivasan, 1997]. This provides a 360^o panoramic visual field in the horizontal plane and 120^o in the vertical plane, around the

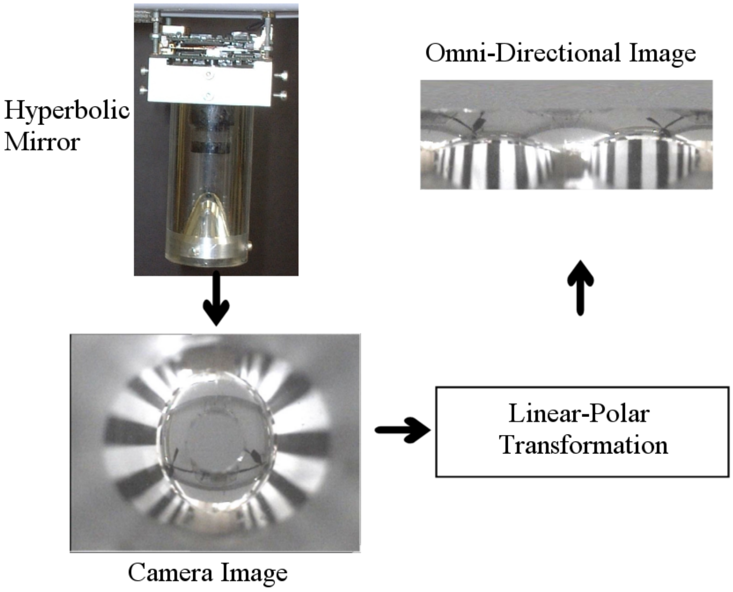

Figure 3.2: *The omni-directional camera. A CCD camera pointed at a hyperbolic mirror provides the panoramic view, a* 360^o *visual field in the horizontal plane and* 120^o *in the vertical plane. During our experiments, the camera image has a resolution of 240×180 pixels. The linear-polar transform transforms the camera image of the mirror to an omni-directional image with a resolution of 240×90 pixels. The linear-polar transform is explained in the text and in Figure 3.3b.*

horizon. Just like a bee this is a mono-vision system that can look all around.

Figure 3.3a shows the reflection of light in the hyperbolic mirror. Obviously, the raw image from the mirror is a deformed image of the environment (see the camera image in Figure 3.2). With a linear-polar transform, the camera image can be transformed to an omni-directional rectangular image. Figure 3.3b shows the transform from the camera image into polar coordinates. Circles in the camera image become straight lines in polar coordinates. Point close to the center of the camera image of the mirror, are high up in the visual field, thus appear high in the image in polar coordinates. Point further from the center of the camera image are lower in the visual field. The transform:

$$r = \sqrt{(x_{cam} - \mu_x)^2 + (y_{cam} - \mu_y)^2} \tag{3.1}$$

$$\theta = \arctan\left(\frac{y_{cam} - \mu_y}{x_{cam} - \mu_x}\right) \tag{3.2}$$

Where r is the radius and θ is the angle. The maximum radius is $r = \mu_y$. x_{cam} and y_{cam} are the coordinates in the camera image and $(\mu_x,\, \mu_y)$ is the center of the camera image. The calculation of the arc tangent is normally performed by the `atan2()` function, to determine the quadrant of the result.

The obtained polar coordinates, can be transformed to the omni-directional image by taken the shape of the mirror into account. The used hyperbolic mirror is shaped in such a way

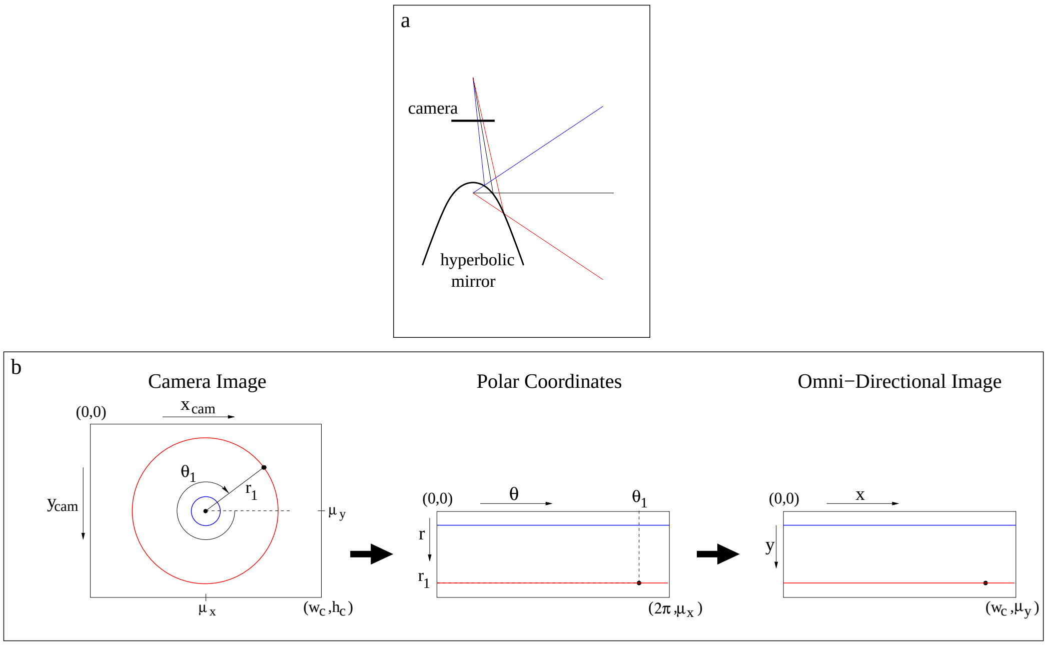

Figure 3.3: **(a)** *shows the reflection of light in the hyperbolic mirror. The mirror is shaped in such a way that each pixel spans the same angle in the visual field.* **(b)** *shows the transform from the camera image of the mirror to the omni-directional image. A polar transform is used to transform the image into polar coordinates and linear transform transforms the polar coordinates to the omni-directional image.* x_{cam} *and* y_{cam} *are the coordinates in the camera image,* θ *and* r *are the polar coordinates and* x *and* y *are the coordinates in the omni-directional image. A circle close to the center of the camera image* (μ_x,μ_y) *is a straight line in the omni-directional image, high up in the visual field (blue). A circle further from the center (red) corresponds to a line lower in the visual field.*

that it is equi-angular. This implicates that each pixel in the camera image spans the same angle of view, irrespective of its distance from the center of the image. (See [Chahl and Srinivasan, 1997] for more detail). The consequence of the equi-angular property of the mirror is that the omni-directional image can be obtained by performing a linear transform of the polar-coordinates.

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{w_c}{2\pi} \\ 1 \end{pmatrix} \cdot \begin{pmatrix} \theta \\ r \end{pmatrix} \tag{3.3}$$

Where x and y are the coordinates in the omni-directional image and w_c is the width of the camera image ($w_c = 2\mu_x$). The resolution of the original camera image is $w_c \times h_c$. Due to the transformations (3.1), (3.2) and (3.3), the resolution of the omni-directional image is $w_c \times \mu_y$.

During our experiment we used a resolution of the camera image of 240×180 pixels. This gives an omni-directional image with a resolution of 240×90 pixels.

3.1.2 Action

The flying robot can act in three dimensional space. By means of three motors, the robot can translate back and forward, rotate to the left and the right and translate up and down.

There are two propellers at the left and right side of the gondola. Both run at the same variable speed either clockwise or counterclockwise and both are attached to the same spindle, which can change the orientation of the propellers, so that the robot can go up-down and back-forward (see Figure 3.4 A). The rotation of the robot to the left and the right is provided by a single propeller at the tail of the blimp.

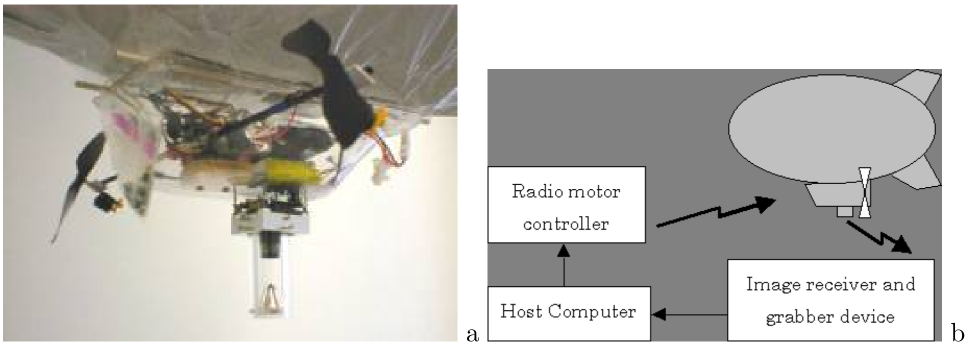

Figure 3.4: (**a**)*The gondola under the blimp. The two propellers are attached to one spindle, which can change the orientation. In this way, the robot can move up-down and back-forward. At the tail of the robot (not shown) is a single propeller, which can let the robot rotate left and right.* (**b**) *shows the sensory-motor loop. The robot sends the images to the host computer. The host computer sends appropriate action commands to the robot, which acts accordingly. This action will make a change in the perception of the robot.*

3.1.3 The Sensory-Motor Loop

The video signal of the camera is sent by wireless transmission to a receiver which is attached to a frame grabber on the host computer, with a maximum frame rate of 25 Hz. At the host computer the processing of the camera image is done and the appropriate action is determined. Three bytes are sent to a digital/analog converter(DAC), one for each movement (i.e., back-forward(BF), rotating left-right(LR) and up-down(UD)). The bytes have values between 0-255. Thereupon, the bytes are converted to voltages between 2-4 Volts and placed on separate channels, which are sent to the radio transmitter. The radio transmitter converts the electric signals to radio signals on different frequencies and sends those to the blimp. The radio signals are received in the blimp, where the different channels BF, LR and UD are interpreted and the actions are performed by the motors. (See Figure 3.4 B.)

During our experiments (see section 4), we obtained a frame rate of 10 Hz. Every second, 10 sensory-motor loops were completed in the algorithm.

3.2 The Elementary Motion Detector

As we discussed in chapter 2, the problem of selecting nearby landmarks based on the apparent speed of the objects in the visual field, consists of two systems: Stabilizing the flight and detecting the angular velocity of objects. In nature, both systems rely on a system that can detect motion in the visual field. This means that in order to make a model for the selection of nearby landmarks, we need a model for detecting motion in the visual field, a so called *motion detector*.

In past research, many motion detectors were proposed, some of them from a statistical or mathematical point of view , and others were a result of biological studies (see [Barron, Fleet, and Beauchemin, 1994] for an overview of motion detectors). There exist good models within both approaches, but since we are working in the framework of biorobotics, we are interested in the biologically plausible models for detecting motion only.

One of the most well known biologically plausible motion detection models is the *Elementary Motion Detector* (EMD), initially proposed by Reichardt [1969]. He performed a series of behavioral studies on the optomotor response of insects, that is, their evoked response to movements relative to themselves in their visual surroundings. He performed these studies to find some fundamental functional principles of the insect central-nervous system, responsible for the optomotor response. Guided by these principles, Reichardt proposed his minimal model for optomotor movement perception, the EMD model. For our model we used the EMD model with some small modifications, as proposed in [Borst and Egelhaaf, 1993], which we will discuss in the next section.

3.2.1 General EMD Model

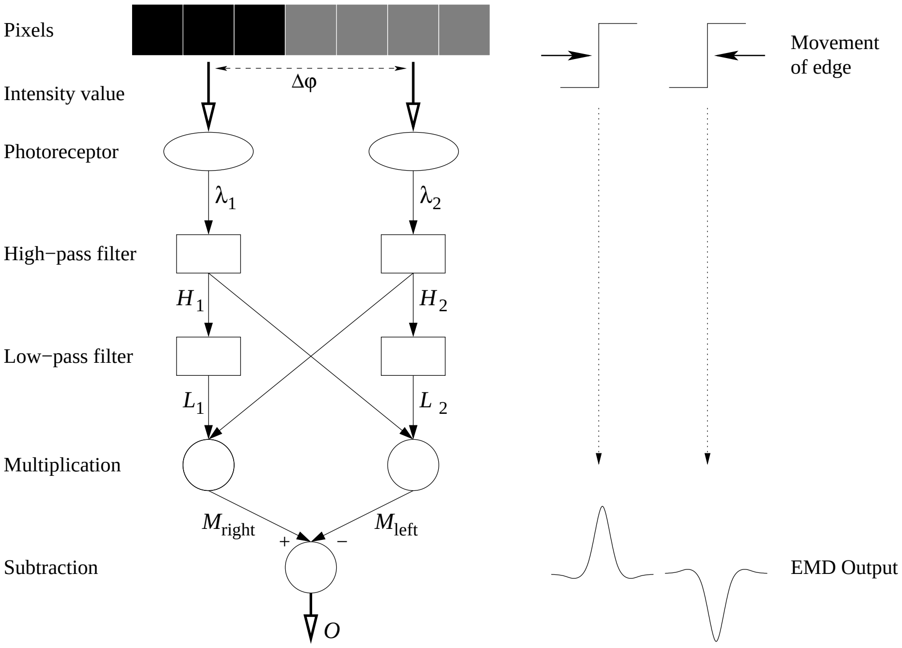

Figure 3.5: *The layout of the EMD-cell as proposed by [Borst and Egelhaaf, 1993].*

Figure 3.5 shows the layout and functionality of an EMD-cell. The input of the cell is provided by two photoreceptors, which receive the luminance or intensity value of two pixels in the camera image, with a distance between them of $\Delta\varphi$ pixels. Both signals are filtered by a first-order difference 'high-pass' filter:

$$\mathcal{H}_i[t] = \lambda_i[\mathrm{t}] - \lambda_i[\mathrm{t} - 1] \tag{3.4}$$

Where λ_i is the luminance (or intensity) as measured by photoreceptor i and $\mathcal{H}_i$ is the output signal of high-pass filter i. This means that the output of the high-pass filter is equal to the change in intensity from time $t - 1$ to time t.
The output of both high-pass filters are then low-pass filtered by a first-order recursive 'low-pass' filter:

$$\mathcal{L}_i[t] = \alpha \cdot \mathcal{H}_i[t] + (1 - \alpha) \cdot \mathcal{L}_i[\mathrm{t} - 1] \tag{3.5}$$

Where LPH_i is the output signal of low-pass filter i and α is the discrete RC-filter coefficient. This is a recursive exponential filter, because an input signal of the filter decays exponentially over time. Due to this decaying property and the fact that the high-pass filter only gives a non-zero output if there is a change in the luminance, the output value of the exponential filter tells us something about how long ago there was a luminance event. The lower the output value of the low-pass filter, the longer ago a luminance event took place. α is the time coefficient of the filter. When $\alpha \rightarrow 0$ the decay is slow and the signal over a long time is considered, when $\alpha \rightarrow 1$, the decay is fast and the signal is only considered over a short period of time.

In the next processing stage, the output of the low-pass filter at one side is multiplied with the output of the high-pass filter at the other side of the EMD-cell.

$$\mathcal{M}_{right}[t] = \mathcal{L}_1[t] \cdot \mathcal{H}_2[t] \tag{3.6}$$

$$\mathcal{M}_{left}[t] = \mathcal{L}_2[t] \cdot \mathcal{H}_1[t] \tag{3.7}$$

Where $\mathcal{M}_{right}$ is the output signal of the multiplication, which is sensitive for motion from left to right. $\mathcal{M}_{left}$ is sensitive for motion from right to left.

When an object moves from left to right and the first edge of the object reaches the left photoreceptor, there is a change in luminance λ_1. This results in a non-zero signal $\mathcal{H}_1$, which comes into the first low-pass filter, resulting in a decaying output signal $\mathcal{L}_2$. Since there has not been a change in luminance λ_2, the output $\mathcal{H}_2$ is still zero, so the output of the multiplication $\mathcal{M}_{right}$ is zero. For a certain period of time both high-pass filters give zero output. When the edge of the object reaches the right photoreceptor of the cell, a luminance event takes place at $lambda_2$, which gives a non-zero output of the high-pass filter $\mathcal{H}_2$. Because there is still an output signal of the low-pass filter, $\mathcal{L}_1$, the output of the multiplication, $\mathcal{M}_{right}$, is non-zero, whereas output $\mathcal{M}_{left}$ is still zero. When the second edge of the object passes the EMD-cell, again there is a pulse in the signal $\mathcal{M}_{right}$ and signal $\mathcal{M}P_2$ stays zero. Because the EMD-cell is symmetrical it is obvious that the signal $\mathcal{M}_{left}$ is sensitive for motion from right to left.

The EMD-cell is also called a *correlation model*, because it correlates an edge at one photoreceptor at a certain time with the same edge at the other receptor later in time.

The final step in the EMD-cell is the subtraction of the signal given by the multipliers, to make the cell sensitive to both leftward and rightward movements.

$$\mathcal{O} = \mathcal{M}_{right} - \mathcal{M}_{left} \tag{3.8}$$

Where $\mathcal{O}$ is the effective output signal of the EMD-cell after the subtraction. If there is motion to the left, $\mathcal{M}_{left} > \mathcal{M}_{right}$, which results in a negative output signal $\mathcal{O}$. Movement to the right gives a positive output. Besides that the cell is sensitive to the direction of the motion, it is also sensitive to the motion's velocity. When the velocity is high, the time between the activation of the first photoreceptor and the second is small. Therefore the low-pass filter gives a high output signal, which results in a high output signal of the EMD-cell, $\mathcal{O}$. Whereas with a slower movement, the signal of the low-pass filter is more decayed, which results in a lower EMD output.

There are two parameters in this model that can tune the sensitivity of the EMD-cell; the time coefficient of the low-pass filter, α, and the distance measured in pixels between the two photoreceptors, $\Delta\varphi$. With $\alpha \to 0$, the cell is sensitive to slow motion. Whereas with $\alpha \to 1$, the decay factor of the low-pass filter is fast, consequently the cell is sensitive to fast motion. Since we are working with a camera input which is not continuous but works with discrete images (i.e., frames), the EMD output is also calculated in a discrete manner. This means that the maximum velocity which can be detected by the cell is $\Delta\varphi$ pixels/frame. A faster motion will simply pass both photoreceptors in between two successive frames and therefore will not activate the EMD cell. So a small value for $\Delta\varphi$ will make the cell sensitive to slow motions and a bigger value makes the cell sensitive to faster motions.

There are some other considerations when tuning these two parameters: A slow decay of the low-pass filter (small value for α) has the drawback that it might result in a wrong correlation, a correlation between two different edges. Because the signal decays so slow, the low-pass filter might still give a signal even when a moving edge has already been detected. When a second edge passes the EMD-cell, the previous edge might interfere with the correlation of the new edge. A similar drawback occurs when the distance between the photoreceptors, $\Delta\varphi$, is to large. In that case, one edge might activate the first photoreceptor, while another edge activates the second photoreceptor. This also results in a wrong correlation between two different edges. So α and $\Delta\varphi$ need to be tuned in such a way that the cell is sensitive to the desired motion velocities and that it is least bothered with the drawbacks.

When we use this EMD-cell as a motion detector, the output signal of the cell tells us the direction of the motion and gives a good indication about the velocity of the motion in the image.

3.2.2 EMD3 Model

Although the EMD model, as described in the previous section, is very well-known and widely used, there are some drawbacks: The model is not only sensitive to the direction and velocity of the motion, but also to the spatial structure of the moving object, to the *intensity values* and the *spatial frequency* [Zanker, Srinivasan, and Egelhaaf, 1999]. Since we want to use the model to detect the direction and velocity of objects moving in the image, this is not desirable. We will first have a look at the problems.

Sensitivity to the Intensity Values

The output of the EMD-cell depends on the intensity values. Imagine a red object moving on a black background. Black gives $\lambda_i = 0.0$ and the luminance of red $\lambda_i = 0.33$. If the object passes the cell, the high-pass filters give an output value of 0.33, which results in a certain output. But if a white object ($\lambda = 1.0$) is moving on a black background, the high-pass filter gives a higher output. Because of the multiplication, this gives a much higher output of the EMD-cell (see Figure 3.6).

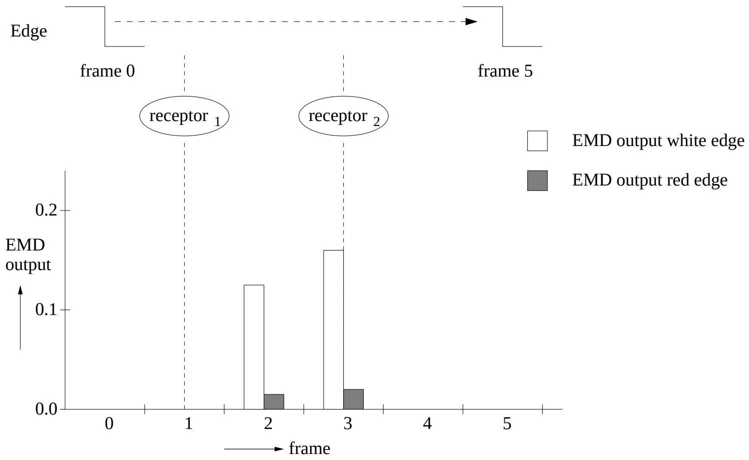

Figure 3.6: *The EMD output at each frame when a white and red edge is moving over a black background both with speed: 1 pixel/frame. The ellipses depict at which frame the edge passes the photoreceptors. Clearly the edge with the highest contrast with the background (the white object) gets a much higher EMD output than the object with less contract with the background.*

So, if there are two objects at the same distance of the robot in front of a black background, one being red and the other white and the robot moves along them. The white object will result in a higher EMD output and will be interpreted as moving faster in the camera image, and consequently will be interpreted as being closer to the robot then the red object.

Sensitivity to the Spatial Frequency

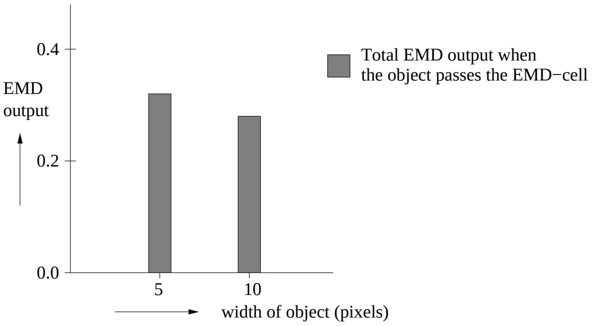

Figure 3.7: *The total EMD output of an EMD-cell with $\Delta\varphi = 2$ pixels when an object passes with velocity: 1 pixel/frame. The output of the EMD-cell is higher when the edges of the object are closer to each other.*

Further more the output of the EMD-cell depends a little on the spatial frequency of the moving object. The smaller the distance between two successive edges, the higher the EMD output (see Figure 3.7. The reason for this is the recursive property of the low-pass filter. When the first edge passes the first photoreceptor of the EMD-cell, the low-pass filter gives a output that slowly decays. When the first edge passes the second receptor, the EMD-cell gives an output, based on the output value of the low-pass filter. But the filter is then not reseted, but there remains a (still decaying) value. When the second edge passes the first receptor, we see, if we take a look at (3.5), that the old output of the low-pass filter is added to the new output of the high-pass filter, which gives a higher new output of the low-pass filter and so a higher output of the EMD-cell. If the two edges are closer to each other, the remaining output value of the low-pass filter is higher, so the output of the EMD-cell is higher.

Solution

We did not solve the last problem of the EMD being sensitive to the spatial frequency of the moving object. A solution to the problem could be to reset the low-pass filter after the matching of an edge between the first and second photoreceptor. Adding this functionality to the EMD-cell would spoil a lot of it's beauty of simplicity. Further more this problem has got a minor impact on the EMD output as we can see from Figure 3.7. So solving the problem is not really necessary.

However, the problem of the EMD being sensitive to the intensity values has got a much bigger impact. This strongly influences the output of the EMD. Different color intensities result in a great difference in the magnitude of the EMD output. To solve this problem, we had to see to it that the high-pass filters give a constant output whenever there is a change in the luminance as sensed by the photoreceptors.

To see to this, a Sigmoid function was placed after the high-pass filter. By doing this, a luminance difference of 0.33 and 1.0, induced respectively by a red and a white object, moving in front of a black background, are both transformed to a signal ≈ 1.0. So when the velocity of both objects is equal, both luminance events give approximately the same EMD output. The drawback of the Sigmoid function is that very small differences in luminance, due to noise in the camera images or small color differences within the object, are enlarged and so result a significant EMD output, whereas without the Sigmoid function the output would be near to zero, as desired. To deal with this drawback, we include a threshold T_λ. This can be described by the following equations:

$$\sigma_i = \begin{cases} \left(\frac{2}{1+e^{-a\cdot\mathcal{H}_i}}\right) - 1, & \mathcal{H}_i \leq -T_\lambda \text{ or } \mathcal{H}_i \geq T_\lambda \\ 0, & -T_\lambda < \mathcal{H}_i < T_\lambda \end{cases} \tag{3.9}$$

Where σ_i is the output signal of the Sigmoid function, T_λ is the threshold and a is the slope parameter. By varying a we can change the slope of the function. At the origin the slope is $\frac{a}{4}$. We want the slope to be steep. (3.9) can then be approximated by:

$$\sigma_i = \begin{cases} -1, & \mathcal{H}_i \leq -T_\lambda \\ 0, & -T_\lambda < \mathcal{H}_i < T_\lambda \\ 1, & \mathcal{H}_i \geq T_\lambda \end{cases} \tag{3.10}$$

The functionality of (3.10) is approximately the same as with the thresholded Sigmoid function (3.9). And since (3.10) is less computational consuming, it is preferable.

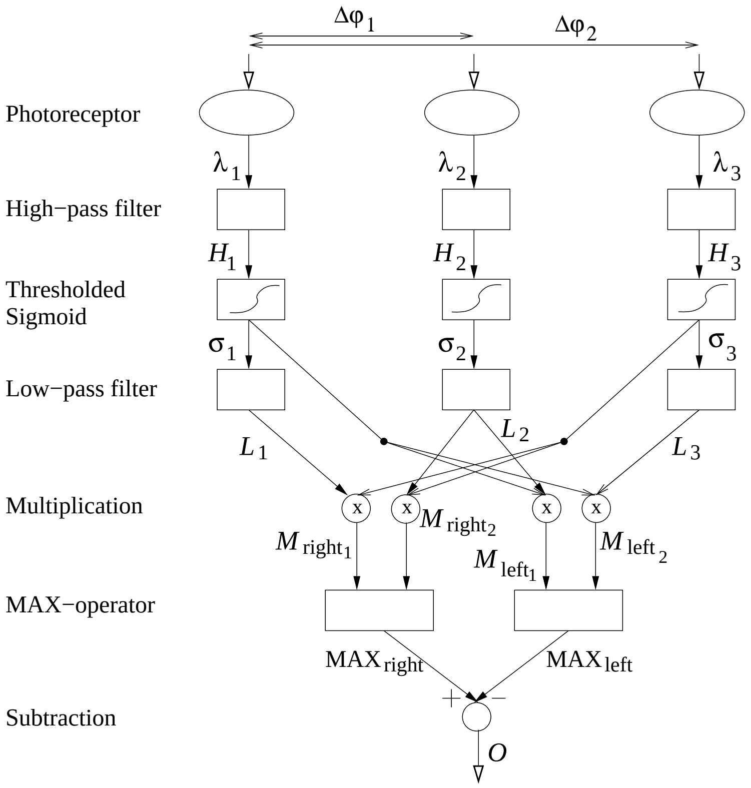

Figure 3.8: *The 3EMD model. In addition to the general EMD model, there is a Sigmoid function after the high-pass filter, which provides a constant output whenever there is a change in luminance λ. And the cell has an extra photoreceptor which allows the cell to be sensitive for slower motions ($v = \Delta\varphi_1$ pix/frame) as well as for faster motions ($v = \Delta\varphi_2$ pix/frame). The MAX operator decides which branch makes the best match. When the widest branch 'wins', the output signal of the multiplier gets multiplied by 2, because the distance between the photoreceptors is twice as much as in the case of the shortest branch. This is the model that is used during our research.*

Figure 3.8 shows that there is another modification to the general EMD-cell. There a third photoreceptor. The reason for this is that the distance between the photoreceptors, $\Delta\varphi$, partly determines to which velocity the EMD-cell is most sensitive. We want the cell to be sensitive to faster motions ($v \approx \Delta\varphi_1$pix/frame) but consequently the cell is than less accurate with slower motions. To add a third 'branch', both the faster ($v = \Delta\varphi_2$pix/frame) and the slower motion ($v = \Delta\varphi_1$pix/frame) can be captured.

With this third branch the multiplications become:

$$\mathcal{M}_{right_1[t]} = \mathcal{L}_1[t] \cdot \sigma_3[t] \tag{3.11}$$

$$\mathcal{M}_{right_2[t]} = \mathcal{L}_2[t] \cdot \sigma_3[t] \tag{3.12}$$

$$\mathcal{M}_{left_1[t]} = \mathcal{L}_2[t] \cdot \sigma_1[t] \tag{3.13}$$

$$\mathcal{M}_{left_2[t]} = \mathcal{L}_3[t] \cdot \sigma_1[t] \tag{3.14}$$

Where $\mathcal{M}_{right_*}$ are the output signals of the multipliers which are sensitive to motion from left to right and the $\mathcal{M}_{right_*}$ signals are sensitive to motion to the left. This EMD layout also demands a decision maker to decide which branch, the wider or the shorter, makes the best match. This decision maker is the MAX-operator which is placed after the multipliers. The branch that has the highest multiplier output signal 'wins' and that signal is passed on. If this is the signal of the wider branch, the signal is multiplied by 2, because the distance between both photoreceptors is twice that of the shorter branch. The MAX operators:

$$\begin{aligned} \text{MAX}_{right} &= \begin{cases} 2 \cdot \mathcal{M}_{right_1}, & \mathcal{M}_{right_1} \geq \mathcal{M}_{right_2} \\ \mathcal{M}_{right_2}, & \mathcal{M}_{right_1} < \mathcal{M}_{right_2} \end{cases} \\ \text{MAX}_{left} &= \begin{cases} 2 \cdot \mathcal{M}_{left_2}, & \mathcal{M}_{left_2} \geq \mathcal{M}_{left_1} \\ \mathcal{M}_{left_1}, & \mathcal{M}_{left_2} < \mathcal{M}_{left_1} \end{cases} \end{aligned} \tag{3.15}$$

Where MAX_{right} is the output of the MAX-operator that is sensitive to motion to the right and MAX_{left} to the left. Finally the EMD output is obtained by:

$$\mathcal{O} = \text{MAX}_{right} - \text{MAX}_{left} \tag{3.16}$$

Parameters

We need to establish a few parameters in order to use the EMD3 model. First of all we need to set the spatial distances in pixels between the photoreceptors (i.e., $\Delta\varphi_1$ and $\Delta\varphi_2$). The maximal detectable angular velocity depends on $\Delta\varphi_2$, for:

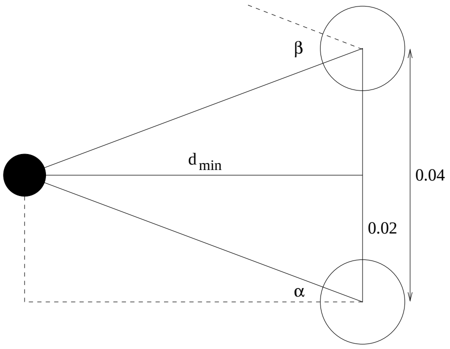

Figure 3.9:

$$\omega_{max} = \Delta\varphi_2 \text{ pixels/frame} \tag{3.17}$$

A few facts about the robot:

Frame rate of camera:	$f = 10$ Hz
Speed of robot:	$v_r = 0.4$ m/s
Resolution of image:	$\rho = 240/2\pi$ pixels/rad

ω_{max} depends on the frame rate of the camera, speed of the robot, resolution of the image and on the minimum distance of an object. The first three variables are facts we have to deal with (see above table). We only have to choose the minimum distance of an object at which it is detectable. An object at a distance closer than that minimum has an angular velocity which is to fast to detect. We choose this minimum:

Minimum distance of object: $d_{min} = 0.5$ m

Now we can determine the maximal angular velocity that should be detectable. Figure 3.17 show the situation when the robot passes an object during an interval of 1 frame. The displacement of the robot in that interval is:

$$x = v_r/f = 0.04 \text{ m} \tag{3.18}$$

The object is at distance d_{min}. We want to know $\angle\beta$ in order to get ω_{max}

$$\alpha = arctan(\frac{\frac{1}{2}x}{d_{min}}) \tag{3.19}$$

$$\beta = 2\alpha = 2 \cdot arctan(\frac{x}{2d_{min}}) = 8.0 \cdot 10^{-2}\text{rad} \tag{3.20}$$

β is the change in angle over a period of 1 frame, so we can say that the angular velocity of the object $8.0 \cdot 10^{-2}$ rad/frame. Since d_{min} is the minimum distance of an object and since the robot passes the object, this is the maximum angular velocity of an object. So:

$$\omega_{max} = \beta = 8.0 \cdot 10^{-2} \text{ rad/frame} \tag{3.21}$$

The camera image is 240 pixels wide and covers 2π rad. This gives a resolution, ρ, of $240/2\pi$ pixels/rad. So,

$$\omega_{max} = \beta \cdot \rho = 3.1 \text{ pixels/frame} \tag{3.22}$$

If we now look at (3.17), we see:

$$\Delta\varphi_2 = \beta \cdot \rho = 3.1 \text{ pixels} \tag{3.23}$$

Since $\Delta\varphi_2$ has to be an integer, we take the smallest integer > 3.1. Because the EMD3 model needs to be symmetrical, $\Delta\varphi_1$ is simply the half of this value:

$$\boxed{\begin{array}{rcl} \Delta\varphi_1 & = & 2 \text{ pixels} \\ \Delta\varphi_2 & = & 4 \text{ pixels} \end{array}}$$

Then we have to establish the time coefficient of the low-pass filter, α in (3.5). The value of this parameter can not be calculated, but has to arise from observations of the model in real-world environments. $\alpha = 0.20$ gave the best results. With higher values the decay of the filter was to fast to detect slower motions. With lower values the problem of a wrong correlation between two different edges, as discussed earlier, came about:

$$\boxed{\alpha \quad = \quad 0.20}$$

The last parameter that has to be established is the threshold in the Sigmoid function, (3.10). This parameter defines how big the difference of luminance should be to mark it as an edge. Again this value is established by observations in the real world. $T_\lambda = 0.10$ turned out to be the most usable. With a lower value, small differences in luminance within one object were wrongly marked as edges. Whereas with a higher value, many edges in the environment were not detected:

$$\boxed{T_\lambda \quad = \quad 0.10}$$

The EMD3 motion detection model as described in this section will form the basis of the *flight-stabilization model* and the *landmark selection model* which we will describe in the next sections.

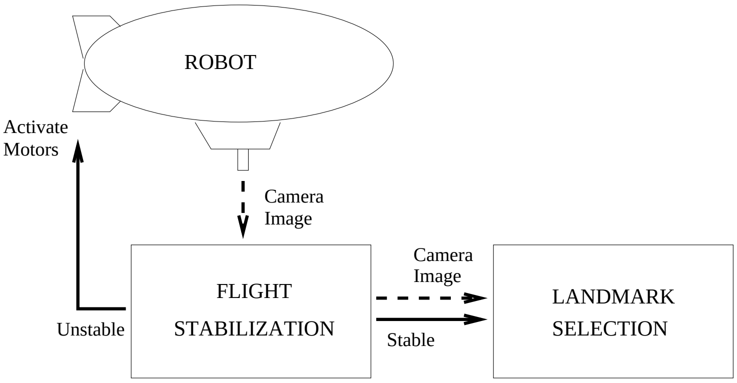

Figure 3.10: *An overview of the complete model. The Flight-Stabilization Model stabilizes the flight and activates the Landmark-Selection Model when the flight is stable.*

3.3 The Flight-Stabilization Model

One of the prerequisites for the model to select landmarks based on image flow, is that the robot has to fly a straight course (see Figure 3.10 for an overview). We concluded in chapter 2 that bees stabilize their flight by using the image motion of the surroundings while they fly. This means that the basis of our *flight-stabilization model* is the motion detection model, the EMD3, that we discussed in the previous section.

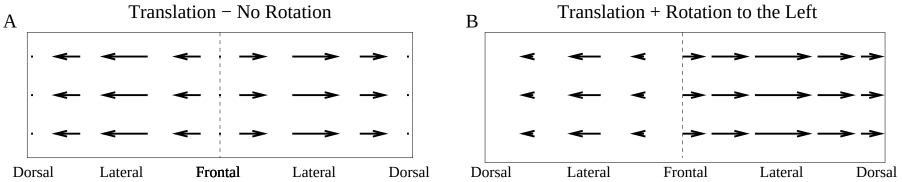

Figure 3.11: *The image flow in different parts of the visual field. The length of the arrow is an indication of the speed of the image flow at that point.* **A** *shows the image flow when the robot is moving translational, without rotation. There is no image flow in the frontal and dorsal part of the visual field. The fastest image flow is in the lateral part. When the robot is rotating there is a change in the image flow pattern.* **B** *shows the image flow when the robot makes a slow turn to the left. Now the speed of the image motion is clearly higher at the right side than at the left side.*

The task of the *flight-stabilization model* is to let the robot fly in one straight line. This means that the model has to see to it that there is no rotation and no elevation of the robot. We do that by using image flow information while moving through the environment. Basically the model tries to equalize the average speed of image motion on the left and right side to stabilize the rotation and on the upper and lower side to stabilize the elevation. If the robot is rotation to the left, the average speed of the image motion is more on the right side than on the left side (see Figure 3.11). The robot than has to turn to the right, to neutralize the leftward rotation. If the robot flies without rotation, the average speed on the left and right side of the robot are equal.

For the stabilization of the rotation we only use the frontal and dorsal (back) part of the visual field. In these the parts there is hardly any image motion when the robot flies straight. When the robot rotates there obviously is a change in the image motion in these parts. Another reason to not use the lateral parts is the following: When the robot flies straight, but close to an object at the left side and far from an object on the right side, the speed of image motion in the left lateral part of the visual field is much higher then at the right lateral part. If we would have used the lateral part to stabilize the course of the robot, the consequence would be that the robot senses a rotation to the right. The frontal and dorsal part are not 'fooled' by the difference between the distance towards objects on the left and right side and so are better used for the stabilization of the robot.

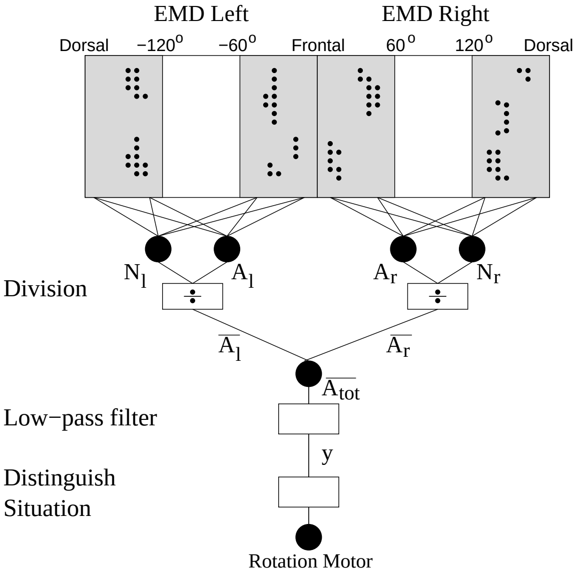

Figure 3.12: *The Flight-Stabilization Model. In this figure, only the model for stabilization of the rotation is shown. The model for stabilization of the elevation is not shown, but is similar. An array of EMD3-cells is placed in the frontal and dorsal part of the visual field on both the left and right side. The cells are horizontally oriented. The output of the active cells on both sides are summoned (A_l and A_r) and the average EMD output of the active cells is taken ($\overline{A_l}$ and $\overline{A_r}$). The rotation of the robot is analysed by looking at $\overline{A_{tot}}$*

Figure 3.12 shows the layout of the stabilization model that we propose. It show only the stabilization of the rotation. This is for simplicity. The stabilization of the elevation is similar, with the only difference that the orientation of the EMD3-cells is different. For rotation stabilization, the cells are oriented horizontally, which means that the three inputs of the cell are obtained from pixels in the same row, with a distance between the pixels of $\Delta\varphi_1 = 2$ pixels. In the case of elevation, the cells are placed vertical, in the same column.

The EMD3-cells are placed in the frontal and dorsal part of the image. For the left side this means that the cells are placed between 0^o and -60^o and between -120^o and -180^o, on the right side between 0^o and 60^o and 120^o and 180^o. In these parts are placed as much cells as possible, that means there is the same number cells as there are pixels, that is $\frac{2}{3} \cdot 240 \cdot 90 = 14400$ cells. The reason for this big number of EMD3-cells is that it makes the model more robust, because the model is less bothered with the noise on the sensory input (i.e., noise on the camera input). The output of all the EMD-cells is calculated each time step, i.e., each new camera frame. When the output is above the threshold T_a, the EMD-cell is active and the output value is passed through. We had to work with this threshold, because the EMD-cells still give a small output, even when an edge passed a long time ago. The reason for this is the recursive property of the low-pass filter (see (3.5)). We only want to use EMD-cells with a significant output. We take the summons of the output of all the active cells on the left side and on the right side. So,

$$A_l = \sum_{i=1}^{N_1} \Delta(\mathcal{O}_i, T_a) \qquad\qquad A_r = \sum_{i=1}^{N_2} \Delta(\mathcal{O}_i, T_a) \tag{3.24}$$

Where A_l is the sum of all the active cells on the left side and A_r on the right side. N_1 is the number of EMD-cells on the left side of the camera image and N_2 on the right side. $\mathcal{O}_i$ is the output of cell i. And the $\Delta(u, \Theta)$ the threshold function:

$$\Delta(u, \Theta) = \begin{cases} 0 & \text{if } u \leq \Theta \\ u & \text{otherwise} \end{cases} \tag{3.25}$$

We used:

$$\boxed{T_a \quad = \quad 0.10}$$

$\mathcal{O}_i$ is an indication for the speed of the moving edges in the image, so A_l and A_r are the totals of the speeds of all the edges in the image. This means that if the speed of the edges is equal on both the left and right side, but there are more edges on the right side, $A_r > A_l$. This is undesirable. What we want is the average speed of the moving edges on the left and right side, in other words the average EMD output of the active cell on the left side, $\overline{A_l}$, and on the right side, $\overline{A_r}$. So,

$$\overline{A_l} = A_l / N_{A_l} \qquad\qquad \overline{A_r} = A_r / N_{A_r} \tag{3.26}$$

Where,

$$N_{A_l} = \sum_{i=1}^{N_1} \Lambda(\mathcal{O}_i, T_a) \qquad\qquad N_{A_r} = \sum_{i=1}^{N_2} \Lambda(\mathcal{O}_i, T_a) \tag{3.27}$$

Where,

$$\Lambda(u, \Theta) = \begin{cases} 0 & \text{if } u \leq \Theta \\ 1 & \text{otherwise} \end{cases} \tag{3.28}$$

The total average EMD output is now,

$$\overline{A_{tot}} = \overline{A_l} + \overline{A_r} \tag{3.29}$$

With this signal we can distinct different situations: If the robot flies purely translational, the average EMD output of the active cells on the left side is equal to that on the right side, except for the difference in sign (i.e., at the left side the motion is to the left, so gives a negative EMD output and on the right side to the right, a positive output). If the robot makes a turn to the left, the average speed of the edges on the right side is more to the right than on the left side to the left. And vice versa with a turn to the right. One problem is that in practice $\overline{A_l}$ and $\overline{A_r}$ are never exactly equal, which means that the robot never thinks it flies straight and is always adjusting its course. The *flight-stabilization model* then would never permit the *landmark-selection model* to select landmarks. Therefore we introduce a range within which the robot is considered to fly straight. This range is defined by T_s. This makes:

$$\begin{array}{ccccl}
 & & \overline{\mathbf{A_{tot}}} & & \textbf{Rotation} \\
\hline
-\overline{A_l} < \overline{A_r} - T_s & \Leftrightarrow & \overline{A_{tot}} > T_s & \Rightarrow & \text{Rotation to the left} \\
-\overline{A_l} > \overline{A_r} + T_s & \Leftrightarrow & \overline{A_{tot}} < -T_s & \Rightarrow & \text{Rotation to the right} \\
 & \mathit{else} & & \Rightarrow & \text{No rotation: Strable flight}
\end{array} \tag{3.30}$$

We used:

$$\boxed{T_s \quad = \quad 0.10}$$

The final step of the *flight-stabilization model* is to let the robot compensate if there is rotation. If there is rotation to the right, the rotation propeller at the tail of the robot is activate, so that the robot turn to the left, in order to neutralize the rotation. The higher $\mid \overline{A_{tot}} \mid$, the more the robot is rotation, so the more force the rotation motor provides to compensate the rotation.

Figure 3.12 shows the overview of the flight stabilization model, with a low-pass filter stage after $\overline{A_{tot}}$ is calculated. The reason for this is again *noise on the camera input*, due to the wireless transmission of the data. The low-pass filter is an average filter:

$$y[t] = \frac{1}{5} \cdot \sum_{\tau=t-4}^{t} \overline{A_{tot}[\tau]} \tag{3.31}$$

Where $y[t]$ is the output signal of the averaging filter and serves as input for the *situation distinguisher*, (3.30).

The model which we described in this section will take care of the stable flight of the robot, which is a precondition to select nearby objects as landmarks based on image flow. In the next section we will describe the model that will select the landmarks.

3.4 The Landmarks Selection Model

We discussed in chapter 2 which objects insects select as landmarks and how they do that: Insects select salient objects close to the goal as landmarks, by using the image motion of the objects to gain distance information. This means that also this model should be based on a motion detection model. So the EMD3 model is also the basis for the landmark-selection model.

Since the robot is flying straight, there is no image flow in the frontal and dorsal part of the visual field and the image flow is the most obvious in the lateral part. This means that we can only gain reliable distance information from image flow in the lateral part of the visual field. For that reason, we used the EMD3-cells in both lateral parts of the camera image. The areas where the cells are located are between -30^o and -150^o horizontal angle on the left side and between 30^o and 150^o on the right side (0^o is the front of the robot). Landmarks are only selected by flying forwards (or possibly backwards) and not by flying up or down. This means that we only use horizontally oriented EMD3-cells. Figure 3.13 shows the layout of the landmark-selection model.

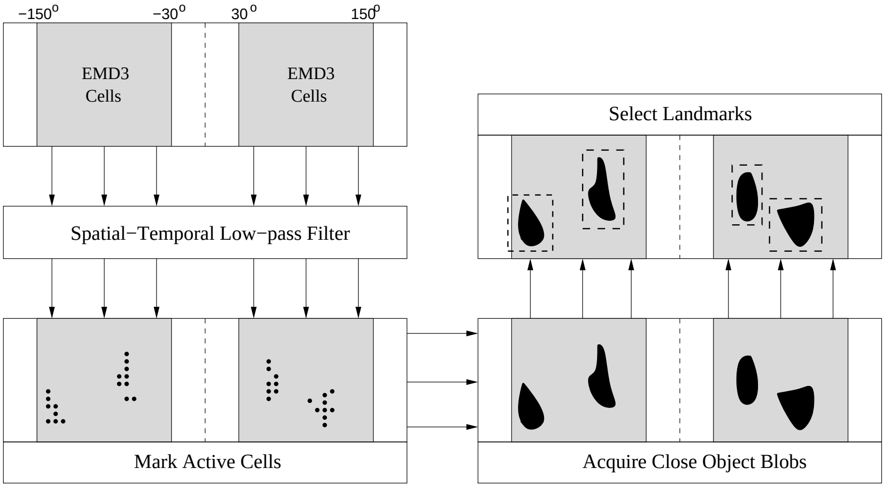

Figure 3.13: *The EMD3-cells are placed in the lateral parts of the visual field, with a horizontal orientation. The outputs of the cells are filtered in both the temporal and the spatial domain. Cells that register image motion that is fast enough (i.e., an object that is close enough) are marked as active cells. These loose active cells form groups. With a clustering algorithm called* blob coloring, *blobs are acquired. These blobs correspond with objects that are close to the robot. So they are selected as landmarks. With the selection, the* size *and* position *of the blobs in the image is saved, together with the average EMD output of the entire blobs which gives distance information about the landmark. At last, the* part of the image *in which the blobs is lain is saved as a PPM-image. This image represents the* color *and* shape *of the landmark.*

The outputs of the EMD-cells are passed through a low-pass filter, which smoothes the signals both in the space and time domain. The reason for this is, as in the flight-stabilization model, *sensory noise*. Smoothing in only the time domain is not sufficient, because the the edges in the image move. So at one time step an edge is at one EMD-cell and the next time step at another. This means that there is no distinction between a change in pixel luminance due to a moving edge and a change due to noise when only filtering in the time domain is used. We want to 'track' the edge during for a short time. This is the reason that filtering in the space domain is added.

The spatial-temporal filter consists of a 2D averaging filter in the spatial domain and an exponential low-pass filter in the time domain:

$$\chi_s[x,y,t] = \frac{1}{25}\left(\sum_{x'=x-2}^{x+2}\sum_{y'=y-2}^{y+2}\mathcal{O}[x',y',t]\right) \tag{3.32}$$

$$\chi_{st}[x,y,t] = \alpha_{st}\cdot\chi_s[x,y,t] + (1-\alpha_{st})\cdot\chi_{st}[x,y,t-1] \tag{3.33}$$

Where, $\chi_s[x,y,t]$, is the output signal of the spatial filter at position (x,y) at time t. The filter is a 2D averaging filter that takes the average of 25 neighbouring EMD3-cells. $\mathcal{O}[x,y,t]$ is the output of the EMD3-cell at position (x,y) at time t. $\chi_{st}[x,y,t]$ is the output signal of the temporal exponential low-pass at position (x,y) at time t, with time coefficient α_{st}.

The signal $\chi_{st}[x,y,t]$ gives an indication about the speed of the image motion at position (x,y) in the image. Since we want to select nearby objects (edges) as landmarks we want to select the objects with a certain minimum angular speed, which means objects that give an EMD output above a certain value. Therefore we introduce a threshold, T_c on $\chi_{st}[x,y,t]$, that defines within which range objects are considered to be close. The cell at position (x,y) is marked active when the output of the spatial-temporal filter is above the threshold:

$$Act[x,y,t] = \begin{cases} 1, & \chi_{st}[x,y,t] > T_c \\ 0, & \text{else} \end{cases} \tag{3.34}$$

When the robot passes an object (edge) that is within the range, it results in many loose active cells around the position of the edge. We want to create a so called *blob* from the active cells that form a group. Because active cells that belong to one group are sometimes not adjacent, all the neighbouring cells of an active cell are marked active as well. In this way groups of adjacent active cells are formed. The cells in such a group have to be labelled equally for they belong to one blob. To achieve this, we use a region filling algorithm called *blob coloring*, which is a recursive algorithm. All the cells that belong to one blob are labelled as one by getting the same color:

```
color=0;
for(y=y1; y<y2; y++) {
    for(x=x1; x<x2; x++) {
        if($Act[x,y,t] \wedge \neg Colored[x,y,t]$) {
            // a new blob
            color++;
            colorBlob(x,y,t,color);
        }
    }
}

void colorBlob(int x, int y, int t, int color) {
    // color pixel x,y and execute colorBlob for all the neighbouring cells
    if(($x1 \leq x \leq x2$) $\wedge$ ($y1 \leq y \leq y2$) $\wedge$ $Act[x,y,t]$) {
        Colored[x,y,t] = true;
        Image[x,y] = color; // All the cells that belong to one blob get the same color
        colorBlob(x-1,y-1,t,color);
        colorBlob(x,y-1,t,color);
        colorBlob(x+1,y-1,t,color);
        colorBlob(x+1,y,t,color);
```

```
            colorBlob(x+1,y+1,t,color);
            colorBlob(x,y+1,t,color);
            colorBlob(x-1,y+1,t,color);
            colorBlob(x-1,y,t,color);
        }
    }
```

So a blob is a group of cells that give an EMD output above the threshold and so correspond with an objects that passes the robot at a short distance. In other words, the blob represents a landmark.

In section 2.2.3 we concluded that bees learn the apparent size of the landmark, the position of the landmark on the retina, some distance information, the color of the landmark and the shape of the landmark. We want our model to learn the same cues about the landmarks. We learn the size by saving the *width* and *height* of the blob. De position on the retina is represented by the position of the *center of the blob* in the camera image. The *average EMD output* of the entire blob gives information about the distance of the landmark. And the color and shape information is learnt by saving that *part of the image* which includes the blob. This gives a small image of the landmark in *PPM format*.

With the saving of this information about the landmark we say that the landmark is selected. The information is required, in order to be able to localize the landmarks during the *landmark navigation* (e.g. as described by the *snapshot model*).

To complete the landmark-selection model, we have to establish the threshold, T_c, which determines the maximum range of 'nearby' objects. We want to classify objects within a range of $2.00m$ as being close, these objects have to be selected as landmarks. Objects at a distance $\geq 2.00m$ should not be selected. Tests with this model implemented on the flying robot result in a threshold that satisfy the requirement:

$$\boxed{T_c \quad = \quad 0.18}$$

The objects that are selected as landmark should have a salient property, otherwise its not possible to find the landmark back. This model will take care of this, since it is inherent to the EMD model: The motion of objects that are not salient (enough), have unclear edges and will not be detected by the EMD3-cells, because of the threshold on the high-pass filter (3.10). Therefor these objects will not be selected as landmarks.

3.5 Conclusion

In this chapter we described a model of the *learning phase* of landmark navigation. In this phase, the landmarks in the unknown environment are learnt, so that the robot is able to return to that location. We proposed a model to select objects close to the robot as landmarks. To be able to create this model we had to make a model that stabilizes the course of the robot. A nice aspect about these models is that they both rely on the same underlying motion detection model, the EMD3 model. In the flight-stabilization model the EMD3-cells in the frontal and dorsal part of the visual field are used whereas the landmark-selection model uses the cells in the lateral parts. The beauty of the underlying EMD3 model is that it can also be used to avoid objects, to regulate the speed and altitude of the robot, to provide a safe landing and for odometry (see section 2.1.3). Implementations of some of these possibilities have been done by [Mura and Franceschini, 1994] (speed and altitude) [Netter and Franceschini, 1999] (altitude) [Srinivasan, Zhang, and Chahl, 2001] (save landing) and [Iida and Lambrinos, 2000; Franceschini, Pichon, and Blanes, 1991](odometry).

So, the underlying EMD model can be reused for different purposes. Being able to reuse a basic circuit is very important for flying agents, since there are normally very strict constraints on the amount of payload the agent can carry. Moreover, reusing the same circuit is an advantage from a computational point of view, since one underlying model is more efficient and requires less computational power. The same principle of reusing basic circuit for different purposes is also found in nature. Lehrer [1998] shows that honeybees use image flow information in different eye regions for different purposes.

In the next chapter we will describe some experiments to test the models and we will show the results of these experiments, with which we can discuss the practicability of the models.

Chapter 4

Experiments

In this chapter we will explain the experiments with the models described in the previous chapter, implemented on the flying robot, *Melissa*. We will show the experimental setup and the results of the experiments with both the *flight-stabilization model* and the *landmark-selection model*.

4.1 The Flight-Stabilization Experiment

The flight-stabilization model as described in section 3.3 is implemented on the flying robot. The experiment in this section is performed, to test whether the model was capable in letting the robot fly a straight course, in such a way that the landmark-selection model can select landmarks based on image flow.

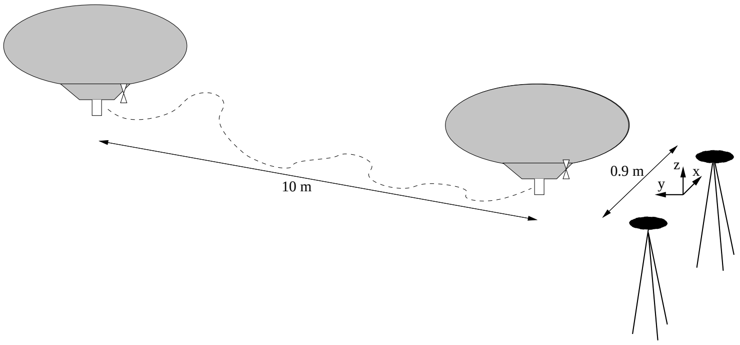

Figure 4.1: *An overview of the experimental setup of the flight-stabilization experiment. The robot flies approximately 10 meters. During that flight the position is observed by two cameras, looking in the direction of the flight. By performing stereo vision, the position of the robot in the 3D space is obtained. This shows the degree of stabilization of the robot's course.*

4.1.1 The Experimental Setup

Figure 4.1 shows the setup of the flight-stabilization experiment. The robot is expected to fly approximately 10 meter in a straight line. The flight of the robot is recorded by two cameras, both looking in the direction of the flight. By means of this stereo vision, the trajectory of the robot in the 3D space can be observed. We obtain the 3D-coordinates as follows:

Stereo Vision to Obtain the Robot's Position

From the two camera images, we want to derive the position of the robot in the 3D space. We will explain the geometry behind this localization task.

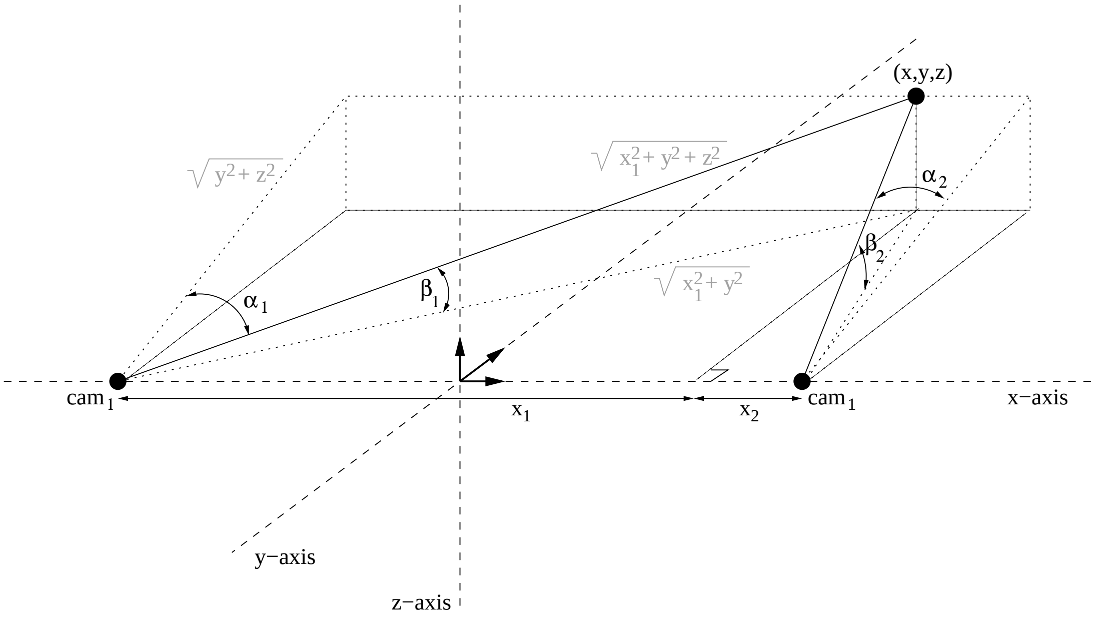

Figure 4.2: *The geometry of the camera setup. α_1 and β_1 are defined by respectively the horizontal and vertical position of the robot in the first camera image and α_2 and β_2 by the position in the second camera image. With these four parameters, the position of the robot, (x, y, z), in the 3D space is derived.*

The distance between cam_1 en cam_2 is 0.9 m. Therefore as we look at Figure 4.2 we see:

$$x_1 + x_2 = 0.9 \tag{4.1}$$

Further more:

$$tan\,\alpha_1 = \frac{x_1}{\sqrt{y^2+z^2}} \tag{4.2}$$

$$tan\,\alpha_2 = \frac{x_2}{\sqrt{y^2+z^2}} \tag{4.3}$$

$$sin\,\alpha_1 = \frac{x_1}{\sqrt{x_1^2+y^2+z^2}} \quad \text{and} \quad sin\,\alpha_2 = \frac{x_2}{\sqrt{x_2^2+y^2+z^2}} \tag{4.4}$$

$$sin\,\beta_1 = \frac{z}{\sqrt{x_1^2+y^2+z^2}} \quad \text{and} \quad sin\,\beta_2 = \frac{z}{\sqrt{x_2^2+y^2+z^2}} \tag{4.5}$$

$$cos\,\beta_1 = \frac{\sqrt{x_1^2+y^2}}{\sqrt{x_1^2+y^2+z^2}} \quad \text{and} \quad cos\,\beta_2 = \frac{\sqrt{x_2^2+y^2}}{\sqrt{x_2^2+y^2+z^2}} \tag{4.6}$$

Combining (4.2) and (4.3) we get:

$$\frac{x_1}{\tan\ \alpha_1} = \frac{x_2}{\tan\ \alpha_2} \tag{4.7}$$

Combining this with (4.1):

$$\begin{aligned} \frac{x_1}{\tan\ \alpha_1} &= \frac{0.9 - x_1}{\tan\ \alpha_2} \\ x_1 \cdot \tan\ \alpha_2 &= 0.9 \cdot \tan\ \alpha_1 - x_1 \cdot \tan\ \alpha_1 \\ x_1(\tan\alpha_1 + \tan\alpha_2) &= 0.9 \cdot \tan\alpha_1 \\ x_1 &= \frac{0.9 \cdot \tan\alpha_1}{\tan\alpha_1 + \tan\alpha_2} \end{aligned} \tag{4.8}$$

Similar to this we can deduce:

$$x_2 = \frac{0.9 \cdot \tan\alpha_2}{\tan\alpha_1 + \tan\alpha_2} \tag{4.9}$$

This gives us the x-coordinate:

$$x = x_1 - 0.45 \quad \text{and} \quad x = 0.45 - x_2 \tag{4.10}$$

If we put together (4.4) and (4.5), we get the z-coordinate:

$$\begin{aligned} \frac{x_1}{\sin\alpha_1} = \frac{z}{\sin\beta_1} \quad &\text{and} \quad \frac{x_2}{\sin\alpha_2} = \frac{z}{\sin\beta_2} \\ z = x_1 \cdot \frac{\sin\beta_1}{\sin\alpha_1} \quad &\text{and} \quad z = x_2 \cdot \frac{\sin\beta_2}{\sin\alpha_2} \end{aligned} \tag{4.11}$$

And finally the y-coordinate by (4.4) and (4.6):

$$\begin{aligned} \frac{\sqrt{x_1^2+y^2}}{\cos\beta_1} = \frac{x_1}{\sin\alpha_1} \quad &\text{and} \quad \frac{\sqrt{x_2^2+y^2}}{\cos\beta_2} = \frac{x_2}{\sin\alpha_2} \\ \sqrt{x_1^2+y^2} = \frac{x_1 \cdot \cos\beta_1}{\sin\alpha_1} \quad &\text{and} \quad \sqrt{x_2^2+y^2} = \frac{x_2 \cdot \cos\beta_2}{\sin\alpha_2} \\ y = \sqrt{\left(\frac{x_1 \cdot \cos\beta_1}{\sin\alpha_1}\right)^2 - x_1^2} \quad &\text{and} \quad y = \sqrt{\left(\frac{x_2 \cdot \cos\beta_2}{\sin\alpha_2}\right)^2 - x_2^2} \end{aligned} \tag{4.12}$$

The parameters in these equations, α_1, β_1, α_2 and β_2, are defined by the horizontal and vertical position of the robot in both camera images.

The cameras have a resolution of 640 × 480. Both cameras have the same horizontal and vertical maximum angle of view. The horizontal angle of view is between $-\theta_1$ and θ_1 and in the vertical direction between θ_2 and $-\theta_2$. The total angle of view is equally divided among all the pixels, so each pixel covers an angle of $2\theta_1/640$ in the horizontal direction and $2\theta_2/480$ vertically.

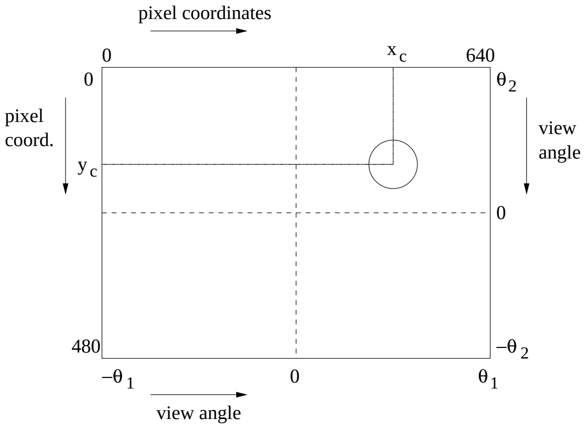

Figure 4.3: *The camera image of the two cameras that are used for stereo vision has a resolution of* 640 × 480 *pixels. The figure show the pixel coordinates on the top and left axes and the associated angles on the bottom and right axes.*

If the position of the robot in the image of camera i measured in pixels is (x_{ci}, y_{ci}), we can derive:

$$\alpha_i = (x_{ci} - 320) \cdot \frac{\theta_1}{320} \tag{4.13}$$

$$\beta_i = (y_{ci} - 240) \cdot -\frac{\theta_2}{240} \tag{4.14}$$

Every 0.5 s the position of the robot is obtained by equations (4.10), (4.11) and (4.12). This results in a series of (x, y, z) coordinates, which represents the flight trajectory of the robot.

The task for the robot in this experiment is to fly a straight course over a distance of 10 meters, using the model described in section 3.3. With this stereo vision setup, the trajectory of the robot during the flights is obtained. In the next section we will show the results of eight trials.

4.1.2 The Results

Figure 4.4 and Figure 4.5 show the result of eight flight-stabilization trials. Figure 4.4 shows the trajectories of the robot in the xy-plane, where the y-axis is in the direction of the flight and on the x-axis are the deviations of the robot to the left and right. So this can been seen as viewed from above. Figure 4.5 gives the view from the side. Here are the deviations of the robot upwards and downwards shown on the z-axis.
If we look at Figure 4.4 and Figure 4.5, we see that, although the robot is flying quite straight during the trials, the robot sometimes slightly deviates from the straight course. This means that the robot is sometimes making a slow turn to the left or to the right or is slowly going up or down. We have to analyse the data to answer the next questions: What is the mean rotational velocity of the robot during these trials? And: Can we consider this as a stable flight, or does the rotation strongly influences the landmark selection task?

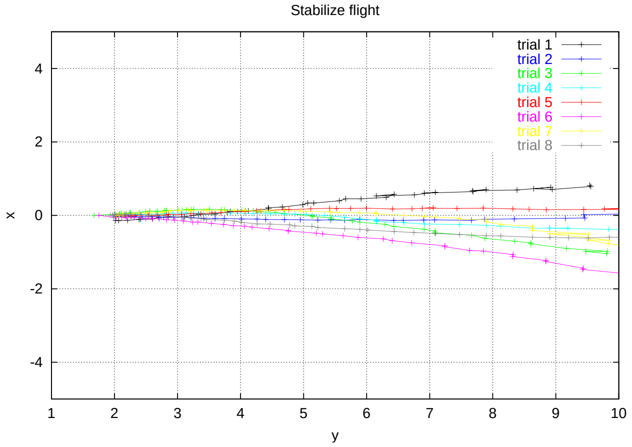

Figure 4.4: *This graph shows the trajectory of the robot during the eight trials in the yx-plane. The y-axis is the direction in which the robot flies. y is the distance in meters. On the x-axis are the deviation to the left and right given, also measured in meters. $x > 0$ means a deviation to the left and $x < 0$ to the right. So we can see this graph as viewed from above.*

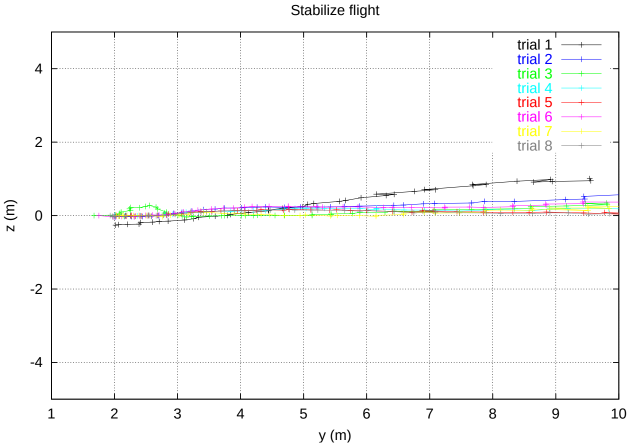

Figure 4.5: *The trajectory of the robot during the same eight trials as viewed from the side. The y-axis is again the direction in which the robot flies and the z-axis gives the deviation of the robot upwards and downwards. $z > 0$ is up and $z < 0$ is down. Both variables are measured in meters.*

Performance analysis

To answer the question what the mean rotational velocity of the robot is during 22 seconds of flight, we have to determine the mean rotation of the robot during the flight. In other words, the difference in orientation at the beginning and at the end of the flight. To calculate this, we use the assumption that the robot rotates with a fixed rotational velocity, which means that the trajectory of the robot describes a part of a circle. The radius of that circle depends on the distance of the flight and the deviation of the robot from the straight course. From

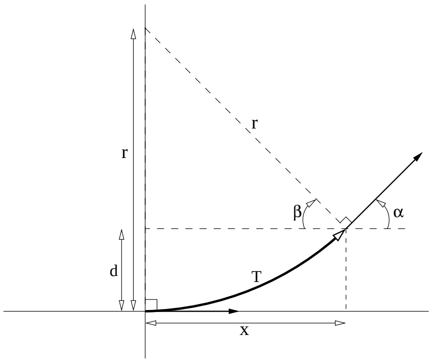

Figure 4.6: *What is the rotation of the robot during the flight? The flight of the robot is approximated by assuming that it rotates with a fixed rotational speed, so the trajectory describes a part of a circle, shown by T. d is the displacement in meters of the robot at the end of the flight and r is the radius of the circle. α is the difference in orientation of the robot between the beginning and the end of the flight.*

Figure 4.6 we see:

$$(r-d)^2 + x^2 = r^2 \qquad \Leftrightarrow \qquad r = \frac{d^2+x^2}{2d} \tag{4.15}$$

where r is the radius of the circle, x is the distance of the flight and d is the deviation at the end of the flight that the robot made. Next, the angle β can be calculated:

$$\beta = \arccos(\frac{x}{r}) \tag{4.16}$$

Finally the the difference in orientation, α, can be calculated:

$$\alpha = \frac{1}{2}\pi - \beta \qquad \Leftrightarrow \qquad \alpha = \frac{1}{2}\pi - \arccos(\frac{x}{\frac{d^2+x^2}{2d}}) \tag{4.17}$$

With this formula and the results of the experiments we can calculate the *yaw* rotation (e.i., the rotation to the left and right). The distance of the flight is $x = 9.0$ m. The mean deviation at the end of the flight is the average over all trials, i, of the absolute deviations, d_i: $d_y = \frac{1}{8}\sum_{i=1}^{8}|d_i| = 0.71$ ($\sigma^2 = 0.24$). We can now calculate the difference in orientation during the whole flight, $\alpha_y = 0.157$ rad. Since the duration of the flight was 22 s, the mean rotational velocity is:

$$\omega_y = 0.157/22 = 7.14 \cdot 10^{-3} \text{ rad/s} \tag{4.18}$$

where ω_y is the rotational velocity of the yaw. This corresponds to $7.14 \cdot 10^{-4}$rad/frame with a frame rate of 10 Hz. And the horizontal displacement of the environment in the camera image due to the rotation, measured in pixels is $\frac{240}{2\pi} \cdot 7.14 \cdot 10^{-4} = 0.027$ pixels/frame.

On a similar way the rotational velocity of the *pitch* (e.i., rotational velocity up and down) can be calculated. The mean deviation in the yz-plane is $d_p = 0.32$ ($\sigma^2 = 0.053$). This gives: $\alpha_p = 0.071$ rad and a mean rotational velocity of the pitch:

$$\omega_p = 3.23 \cdot 10^{-3} \text{ rad/s} \tag{4.19}$$

The height of the camera image is 90 pixels, which covers 100^o or 0.56π rad of the visual field. So ω_p corresponds with a vertical displacement in the camera image of $\frac{90}{0.56\pi} \cdot 3.23 \cdot 10^{-4} = 0.017$ pixels/frame.

Is the observed flight a stable flight? This depends under which assumptions the fight is considered stable, which depends on the purpose of the flight. For instance, the assumption under which the flight of a homing missile is considered stable is when the missile absolutely does not deviate from the straight course. The purpose of the stable flight of our robot is to select relevant landmarks based on image flow. For this purpose, we assume that the flight is stable when the deviation of the robot from the straight course does not strongly influences the functionality of the landmark-selection model.

We pointed out that the robot slightly rotates during the flight. What is the influence of this rotation? When the robot passes an object at 2.00 m, this object has a horizontal angular velocity of 0.20 rad/s. Because of the yaw rotation of the robot this is observed by the camera as an angular velocity between 0.1929 rad/s and 0.2071 rad/s, which corresponds to an object at a distance between 2.07 m and 1.93 m. This is an error range of 3.5%. (The error of objects farther from the robot is slightly more and the error of closer object is less). Since we want to make a separation between *nearby* and *distant* objects, and we do not want to know the exact distance, an error of 3.5% is good enough and absolutely does not influence the functionality of the landmark-selection model.

The pitch rotation is even less than the yaw rotation. Moreover the pitch rotation of the robot works on the vertical axis of the camera image and does not change the horizontal angular velocity of objects, which we use in the landmark-selection model.

In the light of the experiment discussed in this section, we can conclude that the *flight-stabilization model* successfully stabilizes the flight of the robot for the purpose of landmark selection based on image flow. With the stable flight of the robot that this model provides, the landmark-selection model can select landmarks based on image flow.

4.2 The Landmark-Selection Experiment

In this model we discus the experiments with the model for the selection of landmarks, described in section 3.4, that we implemented on the flying robot.

4.2.1 The Experimental Setup

To obtain statistical evidence, an experiment has to be performed a few times under the same conditions. The problem with a freely flying robot is that there are always some differences in the flights of the robot and so the experiments are taken under dissimilar conditions. For that reason we created a rails on which the robot can move (see Figure 4.7), every time exactly the same trajectory, with the same speed and starting at the same moment.

The gondola of the robot moves over the rails with the same speed as the freely flying robot flew during the flight-stabilization experiment. During a 'flight', the gondola passes landmarks, placed at one side of the rail at different distances from the robot. We used only one lateral side out of practical reasons. The other lateral part would give exactly the same

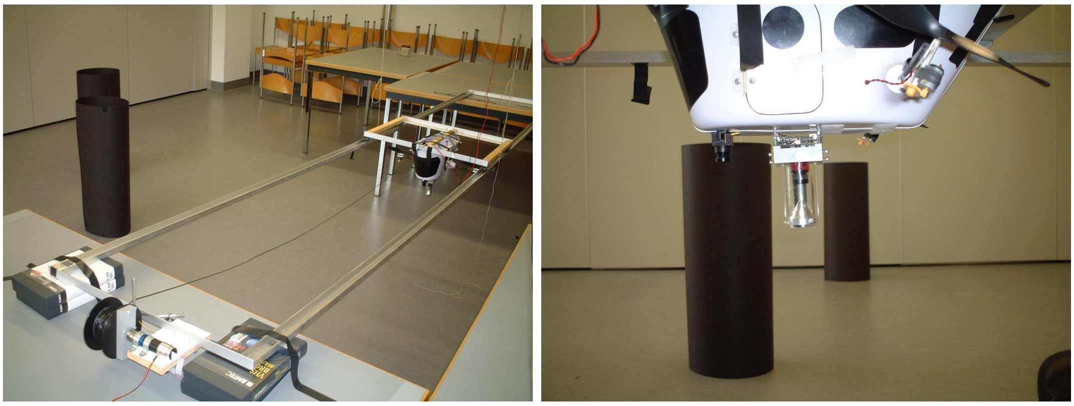

Figure 4.7: *The setup of the landmark-selection experiments. The gondola of the robot moves over a rails, pulled by a motor. The speed of the robot is equal to the speed of the freely flying robot at the flight-stabilization experiment. Landmarks were placed at the side of the rail.*

results, except that the direction of the image flow would be reversed.

In the first experiment a black cylindrical landmark is placed with length 1.00 m and Ø = 0.25 m at different distances from the robot $(0.50, 1.00, 1.50, 2.00, 2.50$ and 3.00 m). The hypothesis is that the model implemented on the robot selects the landmarks within the range of 2.00 m (e.i., at $0.50, 1.00$ and 1.50m) and does not select the landmarks outside that range (e.i., at $2.00, 2.50$ and 3.00m). This hypothesis is discussed in section 3.4.

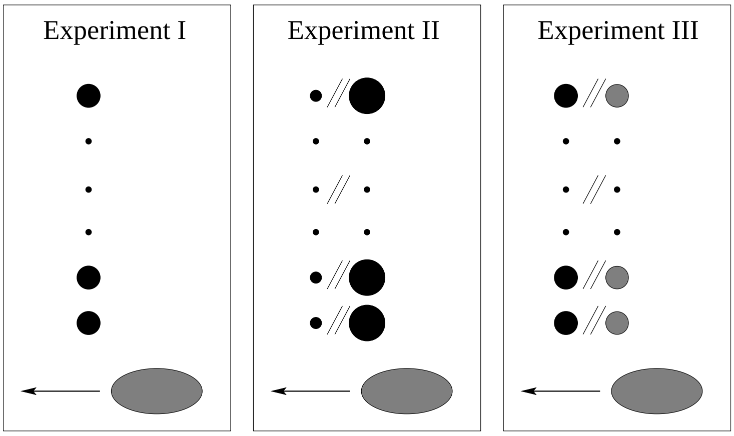

Figure 4.8: *In the first experiment a black landmark of length* 1.00 *m and* Ø = 0.25 *m is placed at* $0.50, 1.00, 1.50, 2.00, 2.50$ *and* 3.00 *m from the robot. Smaller (*Ø = 0.15 *m) and bigger (*Ø = 0.50 *m) black landmarks are used in the second experiment. Again the landmarks are placed on different distances from the robot. In the last experiment a black and a red landmark with* Ø = 0.25 *m are used, also placed on different distances. In all these experiments one landmark per trial is used.*

The distance of the landmark can be obtained in two ways. Based on the apparent speed of the landmark and based on the size. Because in this model the distance has to be obtained by the apparent speed of the object, a second experiment is set up, to see if the *size* of the landmark influences the selection. The experiment is equal to the previous, except that we performed it with a smaller landmark with $\emptyset = 0.15$ m and a bigger landmark with $\emptyset = 0.50$ m. The hypothesis is that the model is invariant to the size of the landmarks and will select those landmarks that are within the range of 2.00 m.

Does the *color* of the landmarks matter? In section 3.2.2 we changed the EMD model so that it should be invariant to the color of object. In this third experiment we test if the color of the landmark really does not influences the landmark selection model. Again the experiment is identical to the previous, but now we compared a black landmark with $\emptyset = 0.25$ m with a red landmark of the same size. Here the hypothesis is that the model is invariant to the color of the landmarks and will select those landmarks that are within the range of 2.00 m.

In all the experiments four trials are taken under the same conditions. See Figure 4.8 for an overview of the three experiments.

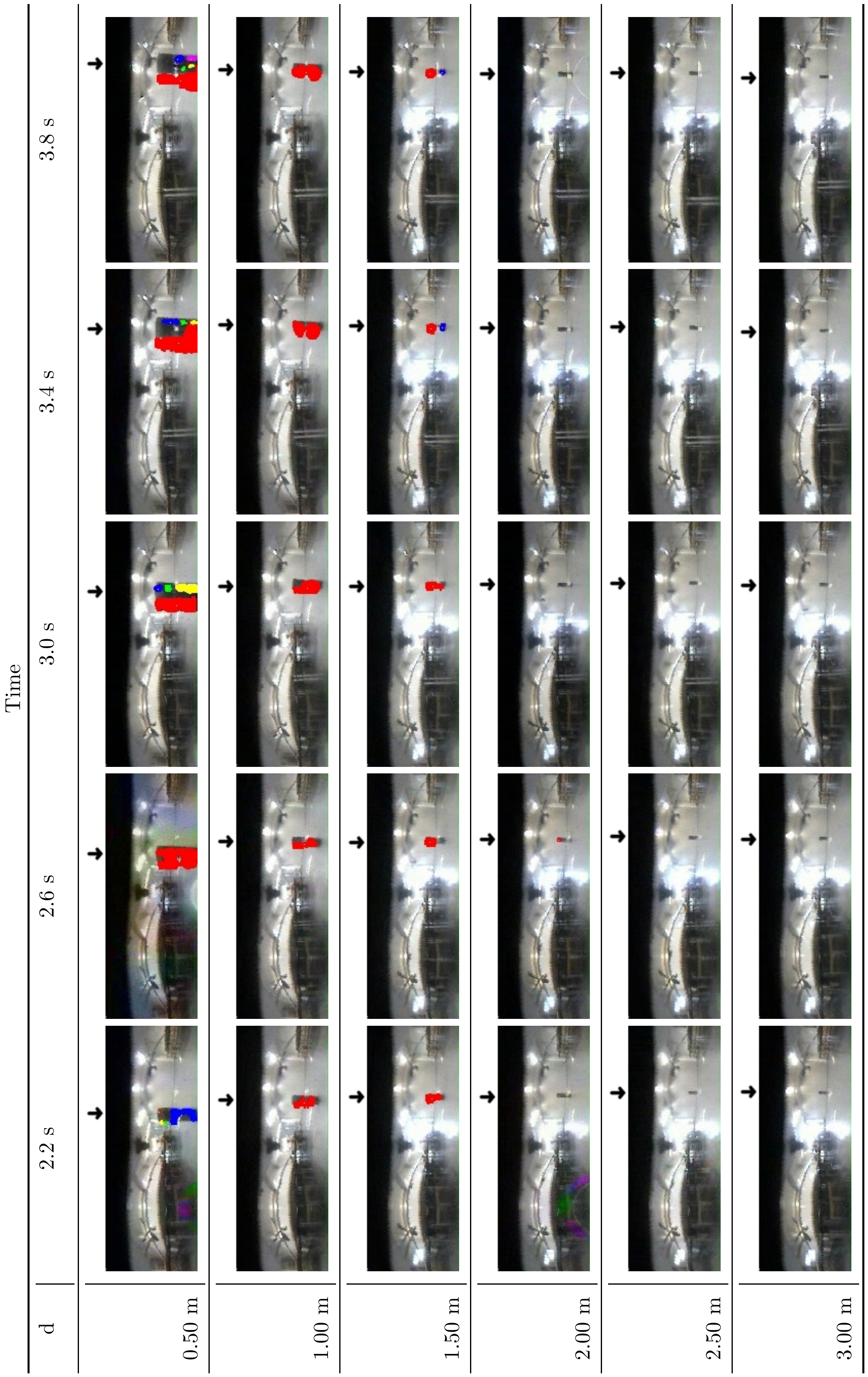

Figure 4.9: *The camera images when the robot passes the landmarks. The images at different points in time are placed horizontal. On the vertical axis is the variation in distance from the LM to the robot, d. The colored blobs depict the selected landmarks. The model learns the landmarks by saving those parts of the camera image where the blob is located. De arrows above the image indicate the position of the landmark in the image.*

4.2.2 Experiment I

The result of the first experiment is shown in Figure 4.9. The figure shows the camera images when the robot passes the landmark, at 2.2, 2.6, 3.0, 3.4 and 3.8 seconds from the beginning of the flight, for all distances. The brightly colored areas in the images are the blobs that are obtained with the method described in section 3.4. When a blob is shown, it means that there is a landmark-selected. The part of the camera image where that blob is located is saved as an image in PPM format.

Figure 4.9 clearly show that the object is selected as landmark when it is within the range of 2.00 m and that it is not selected as a landmark when the object is placed at 2.00 m or further. Only In the image with the landmark at distance 2.00 m at 2.6 s, the object is selected as landmark. But since this is a single failure, not a regularity and the object is not selected in succession, it is not considered a landmark.

Figure 4.10 shows the same results, but in a different representation. The time is placed on the horizontal axis. Time is measured here in camera frames, every frame is 0.1 s. On the vertical axis are the smoothed values of the average EMD outputs of all the EMD3-cells in the blobs shown. These values are also the average over four trials. The different lines indicate the different distances of the object from the robot. The graphs of the objects within the range of 2.00 m show high values, which suggests again that the object is selected as landmark. The graph is positioned higher when the object is closer to the robot, which is correct, since the vertical axis gives an approximation of the angular speed of the object.

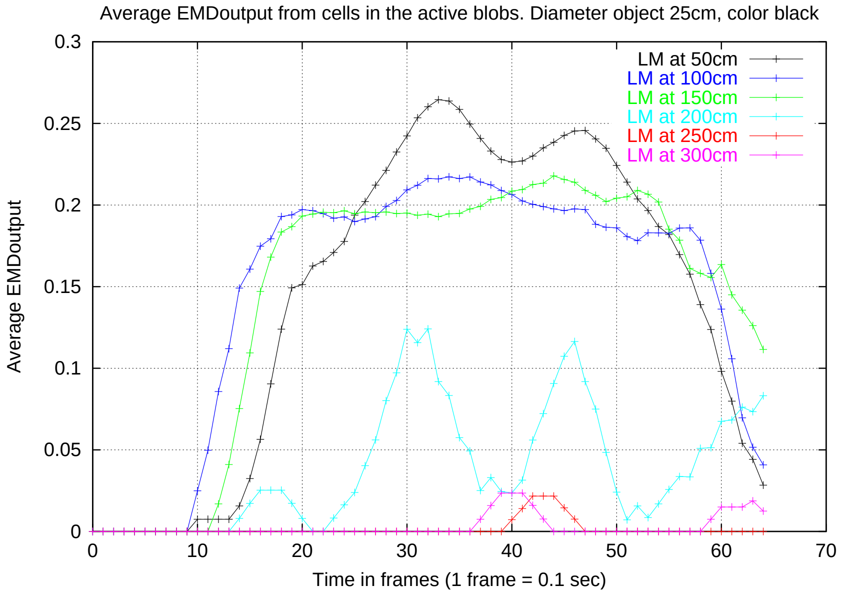

Figure 4.10: *The horizontal axis gives the time measured in camera frames. 1 frame* = 0.1 *s. On the vertical axis is the average EMD output of all the EMD3-cells in the blobs given. This is also the average over four trials. The figure shows six graphs with the object positioned at* $0.50, 1.00, 1.50, 2.00, 2.50$ *and* 3.00 *m from the robot. The object is black with length* 1.00 *m and* $\varnothing = 0.25$ *m. The graphs are smoothed with a moving average filter.*

The graph of the object at 2.00 m is whimsical, which means the object is occasionally selected, but not on a regular basis. Since this is so, the object is not considered to be a landmark. The graphs of the more distant objects show zero output practically all the time. This indicates that the object is hardly ever selected. The reason that these graphs show some values > 0.0 is the result of noise in the camera image.

The graphs have their highest point somewhere in the middle of the time axis. This is according to what we would expect, because the top of the graph is at the moment when the robot is right next to the object, thus where the angular velocity of the object is at his maximum.

4.2.3 Experiment II

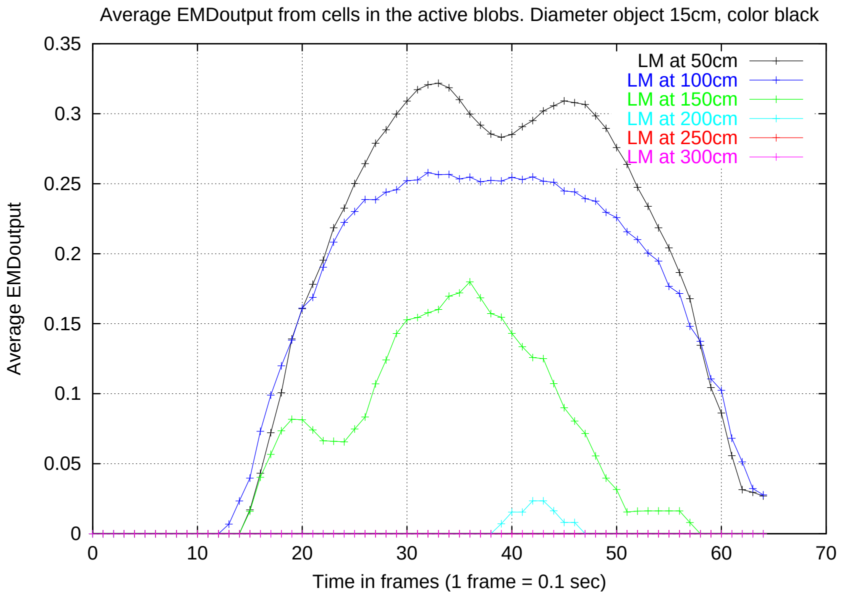

Figure 4.11: *The average EMD output of all the EMD3-cells in the blobs, averaged over four trials. The object used is black with length* 1.00 *m and* Ø = 0.15 *m. For more details on the graph see Figure 4.10.*

In the second experiment the hypothesis that the model is invariant to the size of the landmark is tested. This experiment is identical to the first, except that now the model is tested with a smaller (Ø = 0.15 m) and a larger (Ø = 0.50 m) landmark. Figure 4.11 shows the experimental results with the smaller landmark and Figure 4.12 that with the larger landmark.

The graphs in Figure 4.11 of the trials where the object is placed within the range of 2.00 m show that the object is constantly selected as landmark. The graph of the object at 1.50 m shows low values, but is steady and not whimsical, which shows that the object is constantly selected, but with a low average of the EMD3-cells in the blobs. The graphs with the landmarks outside the range clearly indicate that the object is not selected as landmark.

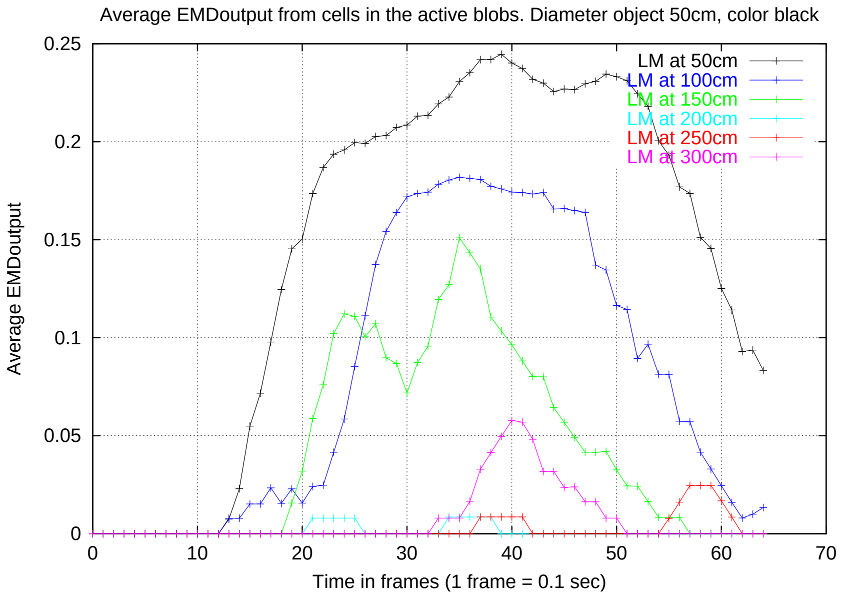

Figure 4.12: *The average EMD output of all the EMD3-cells in the blobs, averaged over four trials. The object used is black with length* 1.00 *m and* Ø = 0.50 *m. For more details on the graph see Figure 4.10.*

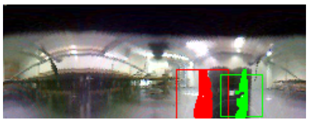

Figure 4.13: *A black object with* Ø = 0.50 *m at* 0.50 *m distance is selected as two landmarks.*

The graphs in Figure 4.12 show the same results as the graphs in the previous figure. The object is selected within the range of 2.00 m and not outside that range. Only the graph with the object at 3.00 m shows an unexpected result. The object is a few times selected, but again not regular enough to label it a landmark.

Figure 4.13 shows how the bigger landmarks are selected when they are close to the robot. Unlike in the case of the smaller landmarks in the previous experiments, the object is now select as two distinct landmark. One at the left edge and one at the right. This is a logical result from the fact that the underlying EMD model basically is a moving edge detector.

From these results we can conclude that the model for landmark selection is not influenced by the size of the object. In both the case of a smaller and a larger object, the model only selects the objects with in the 2.00 m range as landmarks.

Figure 4.14 puts together the graphs of the black object with Ø = 0.15 m, Ø = 0.25 m and Ø = 0.50 m at 0.50 m from the robot. What is notable is that the average EMD output is higher when the size of the object is smaller. We can explain this from the remark in section 3.2.2, that the EMD model is sensitive to the spatial frequency of objects. The smaller an object, the closer its two edges are to each other, the higher the output of the EMD cell when the second edge passes.

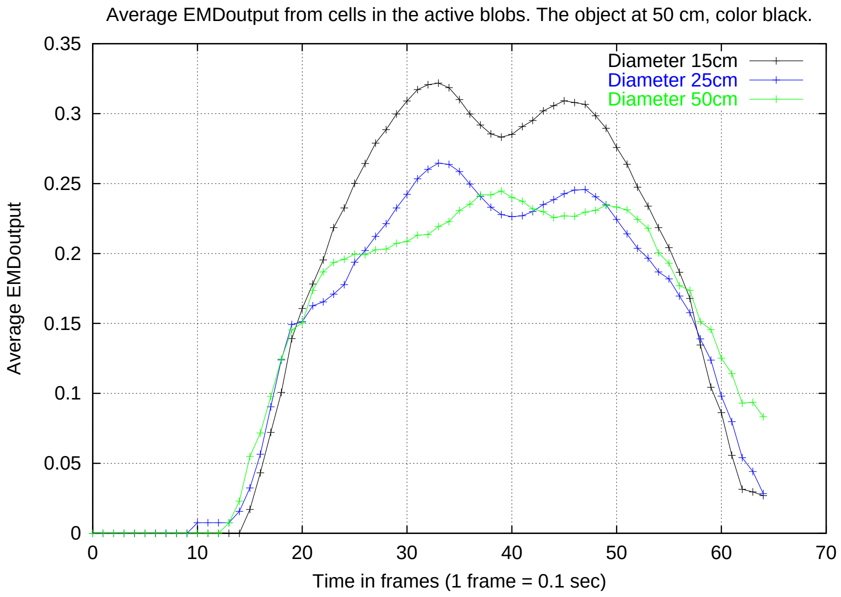

Figure 4.14: *The average EMD output of all the EMD3-cells in the blobs, averaged over four trials. The graphs show the results when using objects of different sizes, with length* 1.00 *m and* $\varnothing = 0.15, 0.25$ *and* 0.50 *m, at* 0.50 *m from the robot.*

4.2.4 Experiment III

In the last experiment, the hypothesis that the landmark-selection model is invariant to the color of the object is tested. A red object of the same size as the black object in experiment I is used. Figure 4.15 shows the results of this experiment. Again we see the same results as in the previous experiments. The nearby objects are selected continuously and the farther objects are not significantly selected as landmarks.

In Figure 4.16, we compare the results of the experiment with the black and red landmark, both of the same size and both at a distance of 0.50 m. What we see is that the top of both graphs is approximately equally high, but that the graph of the red object starts to go up later in time and starts to go down earlier in time. In other words the angle at which the red object is recognizable by the model is smaller than that of the black object. The reason for this is that at a greater angle, the luminance of the red object does not differ enough from the luminance of the background in order for the EMD model to detect the motion. The difference between the two luminances is then below the threshold, T_λ, of equation (3.10).

However, except from the fact that the red object is selected a little later, we can conclude that the landmark-selection model perfectly selects objects of different colors when they are within the range of 2.00 m. The results show that the model is invariant to the color of objects.

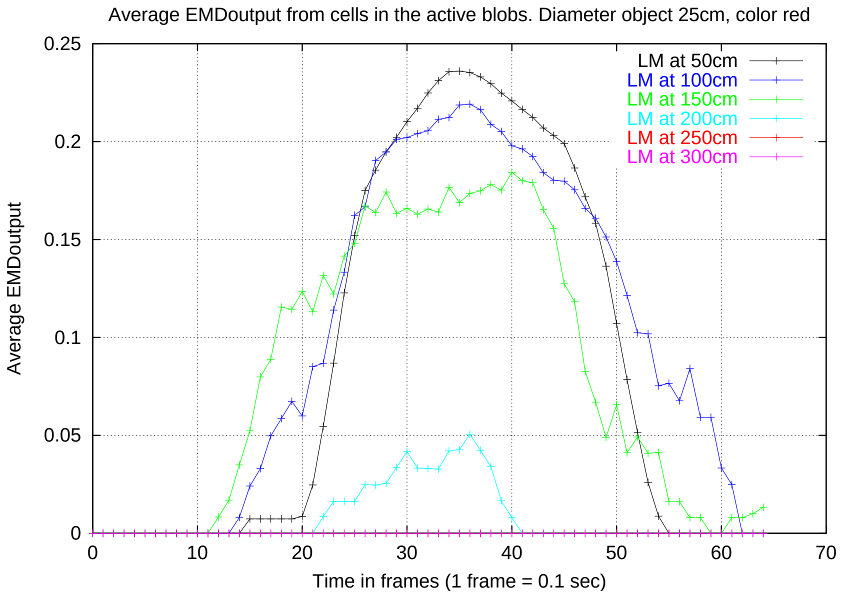

Figure 4.15: *The average EMD output of all the EMD3-cells in the blobs, averaged over four trials. The object used is red with length* 1.00 *m and* Ø = 0.50 *m. For more details on the graph see Figure 4.10.*

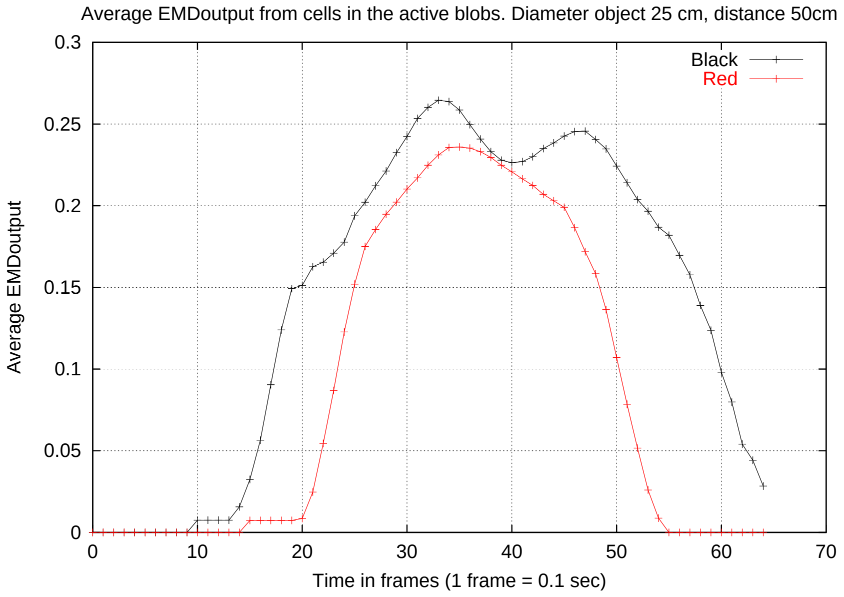

Figure 4.16: *The average EMD output of all the EMD3-cells in the blobs, averaged over four trials. The graph shows the results when using a black and a red object with length* 1.00 *m and* Ø = 0.50 *m, at* 0.50 *m from the robot.*

4.3 Conclusions

In this chapter we performed some experiments with the model of the *learning phase* of landmark navigation, which consists of the two models created in chapter 3. In the first experiment we tested the *course stabilization model* that is based on stabilization mechanisms found in insects. From the results we see that the model is able to stabilize the course of the robot more than sufficient for the *landmark-selection model* to select landmarks based on the image motion of objects. So we can conclude that we created a good working stabilization model, that is based on the same underlying principles as insects use for the stabilization of their course, which we discussed in chapter 2.

The second set of experiments was to test the *landmark-selection model*. As the *flight-stabilization model*, this model is also based on methods adapted from biological studies. The results from these experiments show that the model is able to select nearby objects as landmarks, based on the image flow of the objects, which is produced by the ego-motion of the robot. The underlying EMD model makes a good estimation of the distance towards the objects, and the landmark selection model selects those objects that are within the range of 2.00 m from the robot. The results of the experiments also show that the model is invariant to the size and color of the objects. So we can conclude that we created a working landmark selection model that is biologically plausible.

Chapter 5

Discussion

In this thesis we have attempted to model the *learning phase* of landmark navigation. The model consists of two models; the flight-stabilization model and the landmark-selection model. Both models are the result of biorobotic research. Findings from biological studies on flying insects are used to build these models for the control of the freely flying robot.

The experiments performed with the flight-stabilization model show that we successfully created a biologically plausible model for the stabilization of the course of the robot. The results of the landmark selection experiments show that we also succeeded in building a model that selects landmarks for the use of landmark navigation, in the same manner as honeybees when they learn about a location that they wish to revisit.

A Single Theoretical and Technical Foundation

A nice property of both the flight-stabilization model and the landmark-selection model for robots is that they are both based on the same underlying motion detection model; the EMD model. EMD3-cells in the frontal and dorsal part of the visual field are used for the *stabilization of the flight* and cells in the lateral parts are used for the *selection of nearby objects as landmarks*. Moreover, the EMD model can also serve as the basic circuit for other forms of movement control, such as *controlling the altitude of the robot*, by using the caudal part of the visual field [Netter and Franceschini, 1999; Mura and Franceschini, 1994], *controlling the speed of the robot* by using the caudal, cranial and/or lateral parts [Mura and Franceschini, 1994], *avoiding obstacles* by using frontal part, *obtaining a save landing* by using the caudal part [Srinivasan, Zhang, and Chahl, 2001] and *flying through the middle of a gap* (lateral). Besides that, the EMD model can also be used for *odometry*, by using the caudal, cranial and/or lateral part of the visual field [Iida and Lambrinos, 2000; Franceschini, Pichon, and Blanes, 1991]. See section 2.1.3 for the biological backgrounds.

Being able to use the same basic circuit is an advantage for the *computational power* that is required to control the robot. There is also an advantage when the control of the robot is modeled in analog hardware, like in [Moller, 1999] and [Franceschini, Pichon, and Blanes, 1992]. A shared basic EMD circuit then saves weight in the *amount of payload*, which is a benefit to flying robots.

The EMD itself is a computationally cheap motion detector. In both models we used many EMDs, serially implemented, which obviously slows down the process. However, a nice property of the EMD model is that it is an independent local motion detector, which gives the possibility of parallel processing. If the array of EMDs, used in both models, were implemented as a parallel sensing device, the speed of processing of both models would increase greatly.

Distance Estimator

An important part of the landmark-selection model is that distances towards objects can be estimated, based on the image flow of the objects. This strategy is also found in the locust [Sobel, 1990]. When the locust wants jump on an object, it looks at the object and starts peering, e.g., it moves its head from left to right, while fixating the orientation of the head [Collett, 1978]. By doing this, the locust obtains the motion of the object, which gives information about the distance. This information is used in order to jump on the object. But when the object is moved during the peering, the locust can be fooled. When the object is moved opposite to the movement of the locust's head, the motion of the object is higher and the locust thinks that the object is closer. Therefore the locust jumps in front of the object, instead of on the object. When the object is moved with the movement of the head, but slower, the locust jumps too far. This experiment shows that the locust estimates the distance towards the object, based on the speed of image flow of the object when peering.

This example of distance perception in the locust again shows that the model is biologically plausible, but what is the advantage for roboticists? Why should they use this strategy to obtain the distance towards object, instead of using other techniques like, laser range finding, sonar or stereo-vision? The big advantage of our method is that we use only *one single sensor*, with which we can attain movement control of the robot, selection of the landmarks and navigation on these landmarks, namely the omni-directional camera. Using only one sensor is again a profit for the robot's *amount of payload.*

An advantage of the other methods to gain distance information - like laser, sonar and stereo-vision - is that they are more precise in estimating the distance than our model. But since we only want a rough estimation to know if an object is close or far, our model is accurate enough for the task.

Landmark Selection and the Snapshot Model

We already mentioned that the models proposed in this thesis provide the *learning phase* of the landmark navigation. The models enable the robot to learn an unknown location, so that the robot is able to return to that location.

The ability to return the learnt goal location (i.e., *landmark navigation*) is modeled in the *snapshot model*, which we discussed in section 2.1.2. In [Moller, 2000] a more economical model, based on the snapshot model is proposed, the Average Landmark Vector (ALV) model. Both the snapshot model and the ALV model do not provide the learning phase. If we combine our model with one of these models, we have a complete model that provides the robot with the ability to learn an unknown site and to return to this site by navigating on the learnt landmarks.

Using the Global Positioning System (GPS), would be another option to navigate the robot to a given location. However relying on an external system makes the system not autonomous and it does not conform to the philosophy of biorobotics, since GPS is not biologically plausible. Moreover, such an external system is not everywhere available. Robots on Mars, for instance, can not rely on the GPS system, as yet.

Shortcomings

The model that we proposed has a few shortcomings, which we would like to discuss here.

In the first place is our model not able to select landmarks in the entire surroundings of the goal location. The robot only flies forward and selects the landmarks in the lateral parts of the visual field. To learn the entire environment, the objects in the frontal and dorsal part of the should also be evaluated. During the TBL phase that honeybees perform, the bee solves this problem by hovering from left to right as well as moving backwards. Our robot is not able to hover from left to right because of physical limitations. A different strategy should be used to select landmarks in the frontal and dorsal parts of the visual field.

The second shortcoming is what we discussed before: Our model only provides the learning phase of landmark navigation. It is not able to navigate on the selected landmarks. To realize this, we should couple our model to the snapshot model or the ALV model.

A Related Study

In [Bianco, 1998] and [Lehrer and Bianco, 2000] another model is proposed for automatic landmark selection and navigation for mobile robots, using a TBL-phase: Landmarks are selected by first testing the static and then the dynamic reliability. A landmark is a square of 80×80 pixels in a 640×480 image. Static reliable landmarks are squares that have a high local uniqueness. An equal number of landmarks is chosen from each sector of the visual field. Thereupon the robot performs a TBL phase. It drives from the goal location and moves from left to right, constantly pointing the camera at the goal location. Along the way, the potential landmarks are tracked and their uniqueness is tested. Potential landmarks with an evaluation above a threshold at the end of the TBL are chosen as landmarks. In this way, the model selects objects in the environment that are noticeable, in other words *salient objects*, as landmarks.

During the landmark navigation task, the landmarks are localized in the current view by using a brute force search. The stored images of the landmarks are cross-correlated with all possible parts in the current view, whereby the size of the stored image of the landmark is also enlarged and reduced. The position and the size of the landmark that give the best match is than taken. By comparing this with the size and position of the landmark in the *snapshot*, a tangential and a radial vector is calculated. Adding these vectors of all the landmarks gives a *homing vector*.

This model resembles the model that we proposed in this thesis. Both models are inspired by the TBL phase that bees perform to learn an unknown site. But they differ greatly in which landmarks are selected and how these landmarks are selected. Bianco's model chooses landmarks that are reliable in the sense that they are salient, but the model does not take the distance of the landmarks into account. Landmarks that are salient, but are positioned far from the goal, have a great change to be selected. However, these landmarks are not reliable, since the robot can not use distant landmarks for detailed navigation to the goal location.

Our model does take the distance of landmarks into account. The model proposed in this thesis selects landmarks that are reliable in the sense that they are salient as well as close to the goal. Furthermore our model and the model of Bianco [1998] differ in the strategy how the landmarks are selected. Where we choose a biologically inspired method by using the biologically plausible EMD model to select landmarks based on their image flow, Bianco's model uses classical computer vision methods, which are more computational demanding and far less biologically plausible.

Chapter 6

Conclusion

In chapter 2 we answered the question how insect learn an unknown location in an unknown environment. We have described how bees solve this problem by learning the landmarks in the surroundings. Bees only learn those objects that are salient and near the goal location. They select those objects as landmarks by using the image flow of the objects, induced by egomotion. To gain reliable distance information from the image flow, the flight of the bee during the selection phase need to be stable. In chapter 3 we proposed a model to implement this strategy found in bees on an autonomous flying robot. The model consists of two submodel; a flight-stabilization model and a landmark-selection model. According to the results of the experiments performed in chapter 4, both models that we presented can be called successful. The course stabilization model is able to stabilize the flight of the robot in such a way that the distance information gained from the speed of image flow is reliable. The model for landmark selection successfully selects salient objects that are near the goal location as landmarks, invariant to the size and color of the object.

Future Work

We already pointed out that there are some shortcomings of the model, which are points for future work:

The first point is the fact that the model does not enable the robot to select landmarks in the entire surroundings of the goal. Because of physical limitations, the robot is not able to hover from left to right, and the robot is therefore not able to select landmarks based on the image flow in the frontal and dorsal part of the visual field. Instead of hovering from left to right, the robot could make a 90^o turn and fly straight again. In this way the robot is able to select the landmarks complete surroundings. However, to make exactly a 90^o turn is a very complicated thing to do when the only sensory input is vision provided by the omnidirectional camera. To solve this problem, we should think about using compass information, something that bees also use for their navigation (see [Gould, 1980]).

The second subject for future work is coupling either the snapshot model or the ALV model to our model. The first task to solve is to locate the learnt landmarks in the current view when the robot is displaced from the goal location. The second task to solve is to obtain a displacement vector (or homing vector) from the differences in position and size of the landmarks in the current view and the landmarks as they are saved at the goal location. Finally the robot should be moved according to the displacement vector.

When these two point of future work will be realized, a model is completed that learns a location in an unknown environment and has the ability to return to this location.

Benefits to Biology and Robotics

The presented models are a result of biorobotics research. What is the benefit of this study to both biology and robotics? For biologists, the successful implementation of the models on an autonomous flying robot verifies that the biological theories underlying these models are plausible. Our study may also give biologists more insight in the landmark navigation task of insects. The contribution of this study to robotics is that we presented a model that gives a robot the ability to learn a location in an unknown environment, in order to be able to return to that location. Moreover, the model has the property that it is computationally cheap and that it requires little carrying capacity, for which reason it is possible to implement on a freely flying robot.

References

Barron, J.L., D.J. Fleet, and S.S. Beauchemin. (1994). *Performance of optical flow techniques.*

Bianco, G. (1998). *Biologically inspired visual landmark learning and navigation for mobile robots.* Ph.D. thesis, University of Brescia, Italy.

Borst, A. and M. Egelhaaf. (1993). Detecting visual motion: Theory and models. In F. A. Miles and J. Wallmann, editors, *Visual Motion and its Role in the Stabilization of Gaze.* Elsevier Science, pages 3–27.

Cartwright, B. and T. Collett. (1983). Landmark learning in bees. *Journal of Comparative Physiology A*, 151:521–543.

Chahl, J. S. and M. V. Srinivasan. (1997). Reflective surfaces for panoramic imaging. *Applied Optics*, 36(31):8275–8285.

Cheng, K., T. S. Collett, A. Pickhard, and R. Wehner. (1987). The use of visual landmarks by honeybees: Bees weight landmarks according to their distance from the goal. *Journal of Comparative Physiology A*, 161:469–475.

Collett, T. S. (1978). Peering - a locust behaviour patern for obtaining motion parallax information. *Journal of Experimental Biology*, 76:237–241.

Collett, T. S. and J. Zeil. (1997). The selection and use of landmarks by insects. In M. Lehrer, editor, *Orientation and Communication in Arthropods.* Birkhäuser Verlag Basel, pages 41–66.

Douglass, J. K. and N. J. Strausfeld. (2001). Pathways in dipteran insects for early visual motion processing. In J. M. Zanker and J. Zeil, editors, *Motion Vision - Computational, Neural and Ecological Constraints.* Springer Verlag, Berlin, pages 67–81.

Esch, H. and J. Burns. (1996). Distance estimation by foraging honeybees. *Journal of Experimental Biology*, 199:155–162.

Franceschini, N., J. Pichon, and C. Blanes. (1991). Real time visuomotor control: From flies to robots. In *Fifth Int. Conference on Advanced Robotics: Pisa, Italy.*

Franceschini, N., J. M. Pichon, and C. Blanes. (1992). From insect vision to robot vision. *Phil. Trans. Royal Society of London*, 337:283–294.

Franz, M. O., B. Scholkopf, H. A. Mallot, and H. H. Bulthoff. (1998). Where did i take that snapshot? scenebased homing by image matching. *Biological Cybernetics*, 79:191–202.

Gould, J. L. (1980). The case for magnetic-field sensitivity in birds and bees (such as it is). *American Scientist*, 68:256–267.

Hateren, H.J. van, M. V. Srinivasan, and P. B. Wait. (1990). Pattern recognition in bees: Orientation discrimination. *Journal of Comparative Physiology A*, 167:649–654.

Hausen, K. (1993). Decoding of retinal image flow in insects. In F. A. Miles and J. Wallmann, editors, *Visual Motion and its Role in the Stabilization of Gaze.* Elsevier Science, pages 203–235.

Heisenberg, M. and R. Wolf. (1984). *Vision in Drosophila.* Berlin: Springer Verlag.

Iida, F. and D. Lambrinos. (2000). Navigation in an autonomous flying robot by using a biologically inspired visual odometer. In *Sensor Fusion and Decentralized Control in Robotic System III, Photonics East, Proceeding of SPIE*, volume 4196, pages 86–97.

Lambrinos, D., R. Moller, T. Labhart, R. Pfeifer, and R. Wehner. (2000). A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, 30:9–64.

Lehrer, M. (1993). Why do bees turn back and look? *Journal of Comparative Physiology A*, 172:549–563.

Lehrer, M. (1998). Looking all around: Honeybees use different cues in different eye regions. *Journal of Experimental Biology*, 201:3275–3292.

Lehrer, M. and G. Bianco. (2000). The turn-back-and-look behaviour: bee vesus robot. *Biological Cybernetics*, 83:211–229.

Lehrer, M. and T. S. Collet. (1994). Approaching and departing bees learn different cues to the distance of a landmark. *Journal of Comparative Physiology A*, 175:171–177.

Moller, R. (1999). Visual homing in analog hardware. *International Journal of Neural Systems, special issue: Neuromorphic Engineering*, 9(5):383–389.

Moller, R. (2000). Insect visual homing strategies in a robot with analog processing. *Biological Cybernetics, special issue: Navigation in Biological and Artificial Systems*, 83(3):231–243.

Moller, R., D. Lambrinos, T. Roggendorf, R. Pfeifer, and R. Wehner. (2000). Insect strategies of visual homing in mobile robots. In B. Webb and T. R. Consi, editors, *Biorobotics. Methods and Applications.* MIT Press, Menlo Park, CA.

Mura, F. and N. Franceschini. (1994). Visual control of altitude and speed in a flying agent. In *Proceedings of 3rd international conference on Simulation of Adaptive Behavior: From Animals to Animats III*, pages 91–99.

Netter, T. and N. Franceschini. (1999). Towards uav nap-of-the-earth flight using optical flow. In *ECAL 1999: Lausanne, Switserland*, pages 334–338.

Reichardt, W. (1969). Movement perception in insects. In W. Reichardt, editor, *Processing of Optical Data by Organisms and Machines.* Academic Press, New York, pages 465–493.

Reichardt, W. and T. Poggio. (1976). Visual control of orientation behaviour in the fly. *Quarterly Reviews of Biophysics*, 9(3):311–375.

Ronacher, B., K. Gallizzi, S. Wohlgemuth, and R. Wehner. (2000). Lateral optic flow does not influence distance estimation in the desert ant cataglyphis fortis. *Journal of Experimental Biology*, 203:1113–1121.

Schilstra, C. (1999). *Insect Flight, Eye Movements and Vision.* Ph.D. thesis, Rijksuniversiteit Groningen.

Sobel, E. C. (1990). The locust's use of motion parallax to meadure distance. *Journal of Comparative Physiology A*, 167:579–588.

Srinivasan, M. V., M. Lehrer, S. W. Zhang, and G. A. Horridge. (1989). How honeybees measure their distance from objects of unknown size. *Journal of Comparative Physiology A*, 165:605–613.

Srinivasan, M. V., S. W. Zhang, M. Altwein, and J. Tautz. (2000). Honeybee navigation: Nature and calibration of the "odometer". *Science*, 287:851–853.

Srinivasan, M. V., S. W. Zhang, and N. Bidwell. (1997). Visually mediated odometry in honeybees. *Journal of Experimental Biology*, 200:2513–2522.

Srinivasan, M. V., S. W. Zhang, and J. S. Chahl. (2001). Landing strategies in honeybees, and possible applications to autonomous airborne vehicles. *Biological Bulletin*, 200:216–221.

Srinivasan, M. V., S. W. Zhang, J. S. Chahl, E. Barth, and S. Venkatesh. (2000). How honeybees make grazing landings on flat surfaces. *Biological Cybernetics*, 83:171–183.

Srinivasan, M. V., S. W. Zhang, M. Lehrer, and T. S. Collett. (1996). Honeybee navigation en route to the goal: visual flight control and odometry. *Journal of Experimental Biology*, 199:237–244.

Trullier, O., S. Wiener, A. Berthoz, and J. Meyer. (1997). Biologically-based artificial navigation systems: Review and prospects. *Progress in Neurobiology*, 51:483–544.

Wehner, R. (1982). Himmelsnavigation bei insekten. neurophysiologie und verhal-ten. *Neujahrbl Naturforsch Ges Zurich*, 184:1–132.

Wehner, R., B. Michel, and P. Antonsen. (1996). Visual navigation in insects: Coupling of egocentric and geocentric information. *Journal of Experimental Biology*, 199:129–140.

Zanker, J. M., M. V. Srinivasan, and M. Egelhaaf. (1999). Speed tuning in elementary motion detectors of the correlation type. *Biological Cybernetics*, 80:109–116.