Localisation using Active Mirror Vision System

Luke Cole (u4014181)

Supervised by Dr. David Austin

A thesis submitted in part fulfilment of the degree of
Bachelor of Engineering at
The Department of Engineering of the
Australian National University

June 2006

Except where otherwise indicated, this thesis is my own original work.

Luke Cole
12 June 2006

Typeset in Times by TEX and LATEX 2ε.

To the ongoing struggle to bring robots to our everyday life.

Abstract

Determining the location of the robot in the world is a fundamental problem for mobile robots. This thesis studies the problem of a high-speed vision-based localisation. Two methods have been employed to deal with the problem: a novel vision system and a new technique to be used in the localisation process. The novel vision system uses a parallel mechanism to orient a mirror, which changes the camera view point, instead of the conventional method of re-orienting the camera. This reduces the mass that needs to be re-oriented, which improves the mechanical performance. The new technique developed to incorporate within the localisation process has been termed *view selection* and operates by determining a visual landmark that best improves the localisation approximation and then re-orienting the vision system to that landmark. Experimental results are presented which demonstrate that the robot could localise itself best using one frame of video per second.

Acknowledgements

This thesis was supported by funding from National ICT of Australia and the Australian National University. The Australian National University is funded by the Australian Government's Department of Education. National ICT of Australia is funded by the Australian Government's Department of Communications, Information Technology and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Centre of Excellence Program.

Thank you to my supervisor Dr. David Austin and my brother, Lance Cole for their work on the active mirror vision system. This involved the mechanical hardware design, 3D computer model and machining of the mechanical components. Thank you to the Centre for Sustainable Energy Systems at the Australian National University for creating the mirrors. Thank you to BEC Manufacturing for building the PCB designed in this thesis. Finally, thanks to those who reviewed this thesis.

Indirectly, I would like to personally thank Professor Alex Zelinsky who gave me the opportunity to work with his robotics group, which allowed me to develop my robotic skills at a young age. Furthermore, thank you again to my supervisor and good friend Dr. David Austin who also gave me the opportunity to work with him to continue my robotics endeavors, with research as the primary focus.

Finally, a big thank you to my family for their support throughout the years. If it was not for their guidance and love, this thesis would not exist.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 1920, Karel Capek presented a play *R.U.R. Rossum's Universal Robots* that showed a future in which all workers are automated. A worker was called a *robot*, which is a well known term today as "a mechanical device that sometimes resembles a human and is capable of performing a variety of often complex human tasks on command or by being programmed in advance" (Mifflin 1995). Robots would allow humans to concentrate their concerns on more important tasks, instead of cleaning ones house. Currently the community employees other humans to do their unwanted tasks, however delegation to humans may not be an ideal solution. For some tasks, humans come with major limitations such as a need for sleep, food and a desire for money. On the other hand, a robot does not need to sleep, eat or be paid so the development of robots for the community is promising.

For a robot to perform human-like tasks it must be *mobile* and *autonomous*, not like the robot manipulators (Cox & Wilfong 1990) found in automated car factories. These robots have no sense of the world around them, they only know about what positions their joints are in and what has to be done with them. A mobile robot needs to understand its environment if it is to move around safely, and must be autonomous in operation, since we wish to avoid human supervision. In order to gather information about the environment, sensors such as infrared, ultrasonic, laser and more recently vision are typically used. The amount of visual information present in the real world is clearly immense, so vision presents a very promising sensor. Humans, along with a number of animals use active vision, allowing them to change orientation of their view. There have been a number of active vision systems developed around the world, however there is yet to be a system that is a practical solution due to such issues as physical size and accuracy.

If a robot is going to be mobile it must be able to navigate within its environment, which has been termed *robot navigation*, and is occasionally referred to as the most fundamental problem for robotics (Cox 1991). However robot navigation presents a number of complex problems. One of these problems is known as robot localisation, which is discussed in context in the following Section.

1.1 Robot Navigation

Robot navigation is the task where an autonomous robot moves safely from one location to another. This involves three primary questions (Leonard & Durrant-Whyte 1991):

- *Where am I?* which is known as *robotic localisation*.
- *Where am I going?* which is known as *goal recognition*.
- *How do I get there?* which is known as *path planning*.

To achieve *robot navigation* there are a number of complex issues and difficulties that need to be addressed (Singhal 1997): computation power, object and landmark recognition, obstacle avoidance and multi-modal sensor fusion.

Researchers have often turned to the biological world for inspiration on the robot navigation problem because of its success shown by even the most simplistic insects (Chang & Gaudiano 2000). Studies of ants and bees show that they can localise using simple visual objects (landmarks) within a scene (Fleischer & Troxell 1999, Franz & Mallot 2000). These are inspirational examples given the limited computational resources available.

There have been various approaches to robot localisation (Jensfelt 2001). Vision based localisation is a relatively new approach due to the advances in computing power for the image processing requirements. It is now a popular approach due to its success in the biological world (Salichs 2001).

1.2 Problem Description

The focus of this thesis is to address the mechanical issues and, more importantly, the visual localisation problem. Attempts to solve these areas of concern have so far proved to be impractical, which is due to a number of problems (Singhal 1997), the primary being:

- Slow active vision system and/or algorithm.
- Large size of robot required to carry the vision system.
- Ongoing maintenance of vision system.

When developing an active vision system, design considerations such as size, weight and speed are the major difficulties since a human-like vision system is required. Moving motors and/or cameras in order to change their view point, generally results in poor performance due to the system inertia. Also, a camera and motors require electrical wires, so if a camera or motor needs to be moved, so would the cables causing unwanted wear and bending of the electrical cables, disturbing the tracking of the system. The transmission system of a vision system can also present problems from backlash, if a gearbox is used or high maintenance, if cables are used. Furthermore most active vision system are high in cost. So until an active vision system exists that is small, light and as fast and accurate as our human eyes there will still be room to improve active vision systems.

The problem with visual localisation is that a fast, efficient and reliable system is required. The speed of the system can be hindered by the active vision system itself or the localiser not searching the environment in an intelligent manner, which also affects its efficiency and reliability. The reliability can also be affected by the landmarks used (Panzieri, Pascucci, Setola & Ulivi 2001, Rofer & Jungel 2004, Cassinis, Tampalini & Fedrigotti 2005).

> Effective visual localisers require the solution of two key problems: an active vision system and the localiser. The problems should also be solved in parallel since the performance of the localiser is affected by the active vision system. Hence, the focus of this thesis will be to study the problem of *high-speed vision-based localisation for mobile robots*.

1.3 Approach

In order to achieve a high-speed vision-based localisation system for mobile robots, there are two areas of concern this thesis will study. Firstly, by using an improved mechanism and secondly, by developing a new vision-based localisation method that actively selects a place to look that most improves its localisation. This approach will yield a faster and more reliable active vision-based localistion system.

1.3.1 Active Vision System

The problem with most active vision systems as described above is that the cameras must be oriented. This is a problem because the speed of the system is inversely proportional to the inertia of the cameras and therefore its overall performance. If the cameras could be placed on the fixed platform, along with the motors, the speed of the system could be increased, which would provide faster saccading (rapid eye movement).

This thesis will look at a mirror based approach as a solution for an active vision system, with no orientation of cameras or motors. To achieve this, the cameras will be mounted to a fixed platform and the viewpoint of the system is changed by orienting the mirror itself. The motors will be also mounted to the fixed platform, making it a *parallel mechanism* (*Reminder on Parallel Kinematics* 2003), which is a system where an object is moved via joints to a fixed based as opposed to a serial mechanism such as a manipulator. This reduces the mass that has to be oriented and hence increases the speed and overall performance of the system.

The transmission system will also avoid gear boxes and cable driven mechanisms. This will be achieved through the use of fixed length rods, which will be used to orient the mirror. This avoids backlash, the possibility of cables falling off and reduces ongoing maintenance. Furthermore, the rod transmission system will be considerably cheaper than a gear box or cable driven mechanism.

1.3.2 Localiser Algorithm

The localiser must reliably locate landmarks and do so in a reasonably fast manner. The new style of active vision system, described above, will improve the overall speed of the localiser. However, as mentioned earlier, it will not help if the localiser can not locate landmarks. So, in order to find these landmarks our active vision system must decide on places to look in an intelligent manner.

To achieve this, there are three areas that need to be considered:

- Deciding where to look next in order to locate landmarks for position estimation.
- Recognition of natural landmarks.
- Position estimation of the current location using the landmarks.

1.4 Contributions

Contributions from this thesis on the active mirror vision system were:

- Design requirements for the mechanical mirror vision system.
- Design, implementation and testing of control electronics.
- Development of microcontroller firmware to drive RC servos from RS232 packets.
- Development of host computer software to send RS232 packets to the control electronics.
- Calibration of the CMOS pin-hole camera used.
- Calibration of the parallel mechanism developed.
- Derivation of performance characteristics such as angular velocity, acceleration and accuracy.

The primary contribution of this thesis is the development of the effective localiser, which involved:

- A re-implementation of the SAD landmark detection algorithm (Watman, Austin, Barnes, Overett & Thompson 2004).
- Development of the novel active technique for localisation, which has been termed *view selection*.
- Development of an *IsVisible* algorithm, which determines if a landmark is visible given the robots position, orientation and camera angles.
- Integration into a prior work of Adaptive Monte Carlo Localisation (AMCL), of approximately 8K lines of code.
- Re-implementation of the Graphic User Interface (GUI) output for the MCL (i.e. replaced use of lib-rtk (*RTK Library (deprecated)* 2004), with lib-gtk (*GTK* 2005)).

Finally, the effectiveness of the localiser is demonstrated through experiments. This involves mounting the vision system on a mobile robot and running the developed localisation software within an indoor environment.

1.5 Outline of Thesis

The remainder of this thesis is structured as follows. Chapter 2 gives a detailed discussion on localisation and, more importantly, vision-based localisation. This includes a literature survey of other vision-based localisation systems, along with the main approach to localisation. Chapter 3 discusses the active mirror vision system itself. This includes an overview of the mechanical design and a look into the control hardware and software. Chapter 4 will give a detailed look into the localisation algorithm. This includes an overview of the approach and other algorithms used, in combination with the developed algorithm, on where to look next. Chapter 5 outlines the novel vision system characteristics, along with how they were derived. Chapter 6 presents the experiments and their results that benchmark the developed localisation algorithm. Chapter 7 gives a review of the thesis, highlighting all the main issues of the research, along with discussion of the results and future developments and improvements.

Chapter 2

Localisation Theory

Localisation consists of answering the question *"Where am I?"* from the robot's point of view. That means it is a problem of estimating the robot's *pose* (position, orientation) relative to its environment. The robot's pose is typically the x and y coordinates and heading direction (orientation) of the robot in a global coordinate system.

2.1 Problem Instances

There are a number of problems faced by mobile robot localisation (Thrun, Fox, Burgard & Dellaert 2000). The most simple is the *position tracking problem*. In this problem the robot knows its initial location. Here the goal is to keep track of its current location as the robot navigates through its environment. The solutions to this problem are usually termed *tracking* or *local* techniques (Fox, Burgard & Thrun 1999). A harder problem is known as *global positioning*. Here, the robot does not know its initial location. So it must determine its position from scratch and, therefore, needs to deal with multiple ideas (hypotheses) about its location. The solutions to this problem are called *global* techniques (Fox et al. 1999). The hardest problem is referred to as the *kidnapped robot problem*. This is where the robot knows exactly where it is. But all of a sudden it is 'kidnapped' to a new location. Here the robot needs to determine that it has been kidnapped and, furthermore, must work out its new location. The *global positioning* problem is a special case of the *kidnapped robot problem* where the robot is told it has been kidnapped (Fox et al. 1999).

2.2 Environmental Representation

The environment of the robot is generally represented by a map or a model. Maps can be built in a number of ways and can be either learned by the robot or given to the robot prior to navigation (Singhal 1997):

- **Topological map:** The world is represented as a connected graph. The nodes on the graph indicate important places and the edges indicate paths to each of these important places. Much like a subway map.
- **Feature map:** A geometric map using geometric features to represent the world such as a line map where lines represent the walls of a building.
- **Grid map:** A geometric map were the world is divided into a grid and each cell represents a small area of the world and is given a value to measure the likelihood that cell is occupied.
- **Appearance based methods:** Where the sensor data acquired by the robot is directly mapped to the robot pose.

An example of feature map can be seen in figure 4.3, which is a bird's eye view of the map used in this thesis. It shows a black and white background which represents the valid area (in white) the robot can move within. The small RGB images spread over the image are the features represented by images which are recognisable to the robot.

2.3 Available Information

Simply speaking a robot either *drives* around, or *senses* an environment, which gives rise to the two different kinds of information available: *driving information* gathered by the robot itself as it drives around, secondly, *sensor information* from observations of the environment as the robot moves.

2.3.1 Driving

A *guidance* or *driving* system (Cox & Wilfong 1990) is used to enable the robot to move around with its wheels, tracks or legs. Clearly, the guidance system plays a large role

in the position of the robot, since it directly changes the location of the robot. Without a guidance system the robot could not move around and therefore there is no need for localisation. The guidance system allows the robot to acquire information as it drives around which is known as *relative position measurements*, also called *proprioceptive measurements* (Roumeliotis 1999).

The most obvious relative position measurement is odometry, which comes from the Greek words for *road* and *measure* (Moravec 1998). Odometry is widely used for its short-term accuracy and low cost. For a wheeled robot, encoders are used to measure the odometry by counting the number of revolutions each wheel would do and then integrating over time. However, due to drift and slippage, the integration of the wheel revolutions leads to errors in both distance and orientation (Borenstein 1995), which accumulate over time. Generally errors in orientation cause the largest pose uncertainties. Different sized wheels can also cause errors (Shoval & Borenstein 1995). Odometry is also sensitive to terrain, so if the robot navigates over rocks for example, it may think it has travelled further than it actually has.

Even though odometry causes increasing uncertainty over time, it is rather accurate for short periods of time, and since it is an inexpensive way to gain position information, it is an important source of information for localisation.

2.3.2 Sensing

Sensing information is obtained by acquiring environment information external to the robot, and not information from within the robot. These are known as *absolute position measurements*, also called *exteroceptive measurements* (Roumeliotis 1999). Absolute position information is typically obtained by representing the environment by a set of landmarks such as lines and doors. These landmarks can either be a two-dimensional representation or a three-dimensional representation. When the global position of the landmark is known and an observation of the environment is made, capturing a subset of the landmarks, the robot's position can be determined. Either *artificial* or *natural* landmarks could be used, however, artificial landmarks limit the environment that could be explored. Ultimately a robot would need to learn its landmarks as this would increase flexibility, optimality and autonomy for the robot (Thrun 1998), but that is a problem in itself. However, as will driving information, sensors give errors, which generates uncertainty when matching features.

2.4 Pose Estimation

Estimating the pose is very much about representing uncertain pose information (Fox, Thrun, Burgard & Dellaert 2000). The pose of the robot is often represented by a probability distribution. The distribution may be represented by a Gaussian, in which case, the mean and covariance give the approximation of the robot's pose. The distribution is represented by:

$$p(x_t \mid d_{0\ldots t}) \tag{2.1}$$

Here x is the state, x_t is the state at time t and $d_{0\ldots t}$ are measurements from time 0 up to time t. In robot localisation there are two types of measurements as discussed in Section 2.3. So absolute position measurements will be denoted by a, for *action* and relative position measurements will be denoted by o, for observation, making equation 2.1:

$$p(x_t \mid o_t, a_{t-1}, o_{t-1}, a_{t-2}, \ldots, a_0, o_0) \tag{2.2}$$

Estimating equation 2.2 is the key idea of the Bayesian filtering and is used by most of the pose estimation techniques presented below.

There are many methods to pose estimation, the earliest pose estimation techniques were based on *dead reckoning*, which has been used ever since people started traveling around (Worsley 1978). The following list outlines the possible pose estimation techniques.

- **Dead-Reckoning** Given a initial pose and odometry information allows one to calculate the final pose.
- **Topological:** The uncertainty of the pose information is stored as a graph. The location of the robot is returned as a particular area or room instead of actual coordinates.
- **Gaussian PDF Localisation:** The pose is stored as a Gaussian probability distribution function (PDF).
- **Gaussian Sum PDF Localisation:** When the pose is multi-modal, multiple Gaussian PDF's are used.

- **Position Probability Grids:** The pose space is divided into cells, each representing a possible pose. Thus a discrete approximation of the PDF can be made.
- **Monte Carlo Localisation:** The PDF is represented by a set of samples. Efficient approximation of the PDF is made by finite sized sample set.

Gaussian PDF localisation and Gaussian sum PDF localisation use Kalman filters that are have proved very successful methods for solving the *position tracking* problem (Leonard & Durrant-Whyte 1992), quite effectively solving the problem of compensating for small, incremental errors. Position probability grids use Markov localisation to solve the *global positioning* problem. However this method suffers from two problems: the computational expense, and the fact that accuracy is limited to the resolution of the approximation. Monte Carlo Localisation (MCL) solves the *global positioning* and *kidnapping* problem. It is an approach that overcomes the issues of the Markov localisation by an order of magnitude in its efficiency and accuracy.

2.5 Bayesian Filtering

Particle filters provide solutions to all the problems given in Section 2.1. To understand the implementation of particle filters, let us first derive a *Bayes filter* which is the basis of Markov Localisation. First transform equation (2.2) by Bayes rule (Couch 2001):

$$p(x_t \mid o_t, a_{t-1}, o_{t-1}, a_{t-2}, \ldots, a_0, o_0) = \frac{p(o_t \mid x_t, a_{t-1}, \ldots, o_0)\, p(x_t \mid a_{t-1}, \ldots, o_0)}{p(o_t \mid a_{t-1}, d_{0\ldots t-1})} \tag{2.3}$$

Now a Bayesian filter assumes the environment is *Markov*, that is, if the current state is known, past measurements are independent of future measurements, that is:

$$p(a_t \mid x_t, a_{t-1}, \ldots., o_0) = p(o_t \mid x_t) \tag{2.4}$$

Which simplifies equation (2.3) to:

$$p(x_t \mid o_t, a_{t-1}, o_{t-1}, a_{t-2}, \ldots, a_0, o_0) = \frac{p(o_t \mid x_t)\, p(x_t \mid a_{t-1}, \ldots, o_0)}{p(o_t \mid a_{t-1}, d_{0\ldots t-1})} \tag{2.5}$$

Then the final recursive form is obtained by integrating out the pose x_{t-1} at time $t-1$, yielding from equation 2.5:

$$\frac{p(o_t \mid x_t)}{p(o_t \mid a_{t-1}, d_{0\ldots t-1})} \int p(x_t \mid x_{t-1}, a_{t-1}, \ldots, o_0) p(x_{t-1} \mid a_{t-1}, \ldots, o_0) dx_{t-1} \tag{2.6}$$

The *Markov assumption* also tells us that if x_{t-1} and a_{t-1} are known, the state x_t is conditionally independent of past measurements $o_1, \ldots, o_{t-1}$ and $a_1, \ldots, a_{t-2}$, that is:

$$p(x_t \mid x_{t-1}, a_{t-1}, \ldots., o_0) = p(x_t \mid x_{t-1}, a_{t-1}) \tag{2.7}$$

So equation (2.6) can now be put into recursive form known as *Bayes filter*:

$$\frac{p(o_t \mid x_t)}{p(o_t \mid a_{t-1}, d_{0\ldots t-1})} \int p(x_t \mid x_{t-1}, a_{t-1}) p(x_{t-1} \mid o_{t-1}, a_{t-2}, \ldots, a_0, o_0) dx_{t-1} \tag{2.8}$$

$$= \eta\rho \int \alpha p(x_{t-1} \mid o_{t-1}, a_{t-2}, \ldots, a_0, o_0) dx_{t-1} \tag{2.9}$$

where η equals $p(o_t \mid a_{t-1}, d_{0\ldots t-1})^{-1}$ and is known as the normalising constant, and α and ρ are known as the *motion model* and *perceptual model* respectively. So the equation shows that given α and ρ and the previous approximation of the robots pose $p(x_{t-1} \mid o_{t-1}, a_{t-2}, \ldots, a_0, o_0)$, the current approximation of the robots pose $p(x_t \mid o_t, a_{t-1}, \ldots, a_0, o_0)$ can be determined. Hence, given the initial pose probability distribution the robot pose can always be estimated as the robot travels within its environment.

Its hard to compute the Bayesian filter directly due to issues such as storing the distribution since the probability distribution is continuous and can not integrate that easy. For these reasons particle filters are used, which will be discussed in the following Section.

2.6 Implementation of Particle Filters

The idea of the particle filter algorithm is to represent equation 2.2 by a set of n weighted samples distributed according to the equation 2.2, that is:

$$\{x^i, p^i\}_{i=1,\ldots,n} \tag{2.10}$$

where x^i is a sample (or particle, which is a state) and p^i are called the importance factors, which sum up to one, which makes it a probability distribution, and determine the weight

of each sample.

Before the algorithm is presented the initialisation of the samples will first be discussed in reference to each of the localisation problems mentioned in Section 2.1. In the position tracking problem the initial pose is known, so the initial probability distribution is typically a set of samples distributed in Gaussian fashion around the given initial pose. However in the global localisation problem the initial probability distribution is typically a set of samples uniformly distributed in the state space (environment). The kidnapped robot problem is typically initialised similarly to the global localisation problem, once the robot has determined it was kidnapped.

The particle filter algorithm is generally performed as follows:

1. Sample n particles x^i from the $p(x_{t-1} \mid o_{t-1}, a_{t-2}, \ldots, a_0, o_0)$ according to the importance factors p^i.

2. The robot then moves (acquiring relative position information a_{t-1}). So the samples are moved according to a_{t-1} using the motion model.

3. The robot then makes an observation (acquiring absolute position information o_t), which yields the importance factors and allows the weight for each sample x^i to be calculated using the perceptual model.

4. The new importance factors are then normalised so they sum up to one.

5. Sample new particles from the existing particle set according to the weights. Replace the particle set with the new one. Go to 2.

Figure 6.1 shows an image series of the particle filter in progress. As can be seen, the particle spread is reduced as the robot drives around and makes observations.

2.7 Chapter Summary

This Chapter has discussed the problems with robot localisation, along with the methods for how it is solved. This includes a look at how the robot represents the environment, what information is available to a robot and how the robot interprets this information. The different techniques for pose estimation were also outlined since it is the core of robot localisation. Along with a further discussion on the implementation and theory behind the standard approach to robot localisation, known as the particle filter.

Chapter 3

The Active Mirror Vision System

3.1 Related Work

This Section will outline existing vision systems and show why there is a need for the novel active vision system proposed in this thesis. Generally the active vision system is required to perform human-like eye movements. There are five categories for human eye movements: *saccading*, *smooth pursuit*, *vergence*, *vestibulo-ocular relflex* and *optokinetic response* (Kandel, Schwartz & Jessell 2000). Saccading is rapid eye movement (up to $900^{\circ}.s^{-1}$, or 3 to 4 full range saccades, per second (Kandel et al. 2000)). Smooth pursuit movement is based on maintaining an image of moving object on the *fovea* (Kandel et al. 2000) at speeds below $100^{\circ}.s^{-1}$. The vergence movement is when we adjust our eyes for viewing objects at different depths. The vestibulo-ocular reflex and optokinetic response stabilize the eyes during head movement. Vestibulo-ocular reflex and optokinetic response are important for travelling on uneven terrain, however it is generally not a hardware design requirement because it can be handled in software given the head movement information. Smooth pursuit movement requires high accuracy from the motor control and real-time capability, and high saccade rates requires high speed motors. There have been a number of vision systems designed to perform as well and as fast the human eye, however, there is yet to be a vision system that performs to the standard of the human eye on every level.

In the past the goal for active systems were to experiment with different configurations using large systems with many degrees of freedom (DOF). In 1992 KTH developed an active vision system (Pahlavan & Eklundh 1992) that had 13 DOF, a motorised base-

line and reconfigurable joints. Another system was the Yorick 11-14 (Sharkey, Murray, Vandevelde, Reid & McLauchlan 1993) that had a 55cm baseline (distance between camera) and reconfigurable joints. The trend of vision systems was then to design smaller systems such as the revision of the Yorick head, Yorick 55C (Heuring & Murray 1999), which has an 18cm baseline. One of the design consideration first noted was the use of a common tilt axis for both camera and independent pan axes for each camera, these can be seen in the Yorick models and another vision system known as the PennEyes (Madden & von Seelen 1995).

The next generation of vision system were primarily designed to perform some, or all, of the human-like eye movements discussed above. In 1998 MIT developed a vision system (Scassellati 1998) for the Cog robot (Brooks, Breazeal, Marjanovic, Scassellati & Williamson 1998), which concentrated its design requirements on emulating the human eye movements mentioned above. The interesting design feature of this vision system is the use of four cameras to get a large field of view for each camera, which enables objects to be focused on at different depths simultaneously. The Agile eye (Clement 1996) (see figure 3.1), which was designed so no motor carries the mass of any other motor, achieved a performance better than the human eye, however it is simply too large and heavy and therefore limits the type of robot that could carry the system. Another head that explores this parallel mechanism is the (*EPFL Vision Sphere* 1997) and its stereo version (*EPFL Stereo Vision Sphere* 1998), but they are also rather large, complex and heavy to be used on most mobile robots.

The choice of transmission was another key design factor for the new vision system. The Yorick and the ESCHeR (Kuniyoshi, Kita, Rougeaux & Suehiro 1995) use backlash-free harmonic-drive gearboxes, however this technology causes a large speed reduction ratio that limits the output speed to less than 100rpm (*Harmonic Drive Systems* 2005). In 2000, the CeDAR (Truong 2000) (see figure 3.1) was developed, which has the motors mounted to a fixed platform and orients the camera via a cable driven mechanism instead of conventional gearboxes, which suffer from backlash. However cables can come off or become stretched, which leads to high maintenance since the cables need to be adjusted and/or tightened periodically. A more recent binocular vision system (Tsujita, Konno & Uchiyama 2005) was developed that employs rods instead of cables or gears and rods are also employed in this thesis.

Another solution is simply to use a panoramic camera (Thompson, Matsui & Zelinsky 2000), which would allow one to remove the active component of the system. However,

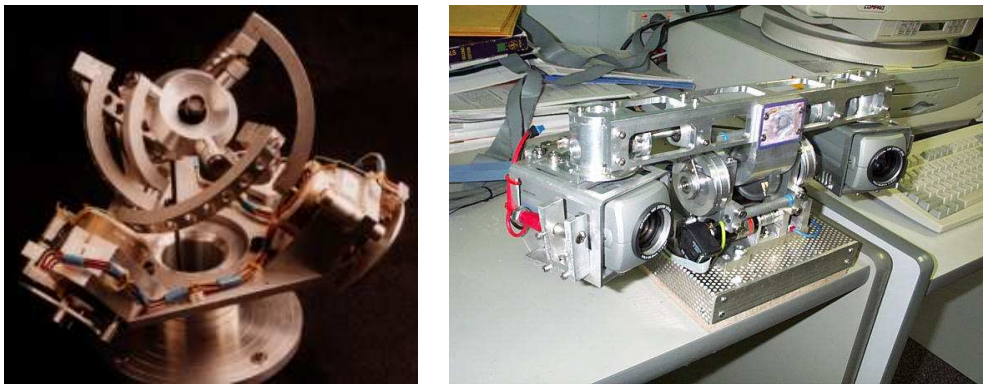


Figure 3.1: Left: Agile Eye, Right: CeDAR

the acquired images have considerably lower resolution, and for a vision-based localisation system that correlates visual features, the low resolution causes significant additional uncertainty.

However the main problem with all these systems is that they are orienting a camera which slows down the system and requires more torque and therefore more power is needed to drive the system. The use of a parallel mechanism over a serial mechanism also presents potential advantages such as: mechanical strength and low moving mass (*Reminder on Parallel Kinematics* 2003). Furthermore, the high cost of most existing active vision systems prohibits their deployment outside of the laboratory.

3.2 Hardware Design

The hardware design process was undertaken by first selecting appropriate hardware to suit the system, this information can then be used to derive the design requirements for the novel vision system.

The camera is the first consideration since the lens size will affect the size of the mirror and the image resolution will affect the resolution required from the motors. Most vision systems use cameras with high resolution, quality and cost, which also tend to have a large lens size, such as the cameras used on the CeDAR (see figure 3.1). Though a CCD pin hole camera would provide the necessary lens size and the cost would be considerably lower, the resolution and quality would be lower. Barrel distortion would also be considerable, however, this can be removed with some computation.

The camera selected was a pin hole camera, since cost and lens size were the primary factors. The datasheet can be found in Appendix A and the primary specifications can be found in Table 3.1 since the key information had to be gathered via e-mail communication. This information enables us to calculate the servo requirements:

$$ServoResolution = \frac{CameraViewingAngle}{ImageHorizontalResolution} \approx 0.094^{\circ} \tag{3.1}$$

$$ServoRangeOfMotion \geq CameraViewingAngle \geq 60^{\circ} \tag{3.2}$$

Once a camera is selected the motor resolution (minimal increment per pixel) can be calculated so the motor can be chosen. A servo motor would be the most suitable here since a DC motor would require a complex gear box to provide the incremental step motion.

RC servo motors were chosen since they are cheap, easily obtainable, easy to control over stepper motors and have a built-in encoder and come with components for the transmission system. The RC servo datasheet can be found in Appendix B and the primary specifications can be found on Table 3.1, again since the key information had to be gathered via e-mail communication. The servo primary specifications show that the resolution and range of motion is well within the requirements given by the camera. The ball-joints used for the rod transmission system were standard accessories for RC servos, the datasheet can also be found in Appendix B.

Given the camera and motor information the final design requirements for the vision system can be can established, which is given in Table 3.1.

The mirror size is determined from the camera viewing angle and the how the silver is deposited. If the silver is deposited on the front of the glass, the mirror can be placed closer to the camera and hence reduces the size of the system. So some mirrors were custom manufactured by Centre for Sustainable Energy Systems at the Australian National University. The mirror was made to a size of $45 \times 65 \times 1$ mm with silver deposited to a thickness of 3.8 K angstroms. The mirror size was determined by the technicians that built this vision system once the design requirements where provided.

The system will be controlled by a computer so a communication standard must be chosen such as USB, RS232 or parallel port control. USB would provide the fastest communication and is also the most modern and popular communication standard. RS232 and parallel would be easier to implement than USB, however it is considerably slower. RS232 requires less wires than parallel, so less pins are needed on a microcontroller and

Camera	
Viewing angle	60°
Image resolution	640×480 pixels
Frame rate	30fps
Lens size	5.2mm
Servo	
Range of motion	120°
Resolution	0.020338°
Angular velocity	$333^\circ.s^{-1}$
Active Mirror Vision System	
Field of view	60°
Range of motion (vertical and horizontal)	60°
Angular resolution	0.09°
Velocity	$600^\circ.s^{-1}$

Table 3.1: Design Requirements

more importantly RS232 is supported better than parallel. So RS232 was chosen since it is a more commonly used standard than parallel and is not as complex as USB.

Figure 3.2 shows the hardware architecture layout. As can the control electronics receives RS232 messages and translates the messages to servo position. So the design of the electronics is a matter of determining how to receive RS232 messages and sending them as PWM signals to the appropriate servos. The RS232 signal voltages (-12V, 12V) can be converted to standard logic voltages (0V, 5V) by using a MAX232 IC, then any microcontroller would be suitable to receive the messages and decode them. A PIC (see (Mircochip 2001) for the datasheet) from Microchip Technology Inc was chosen for its value for money. The PIC chosen has special features to handle RS232 requests so it will make communication more reliable and easier. The electronics schematic can be found in Appendix C, which was designed, assembled and tested as part of this thesis, and the printed circuit board (PCB) was built by BEC Manufacturing. Figure 3.2 shows the hardware architecture layout.

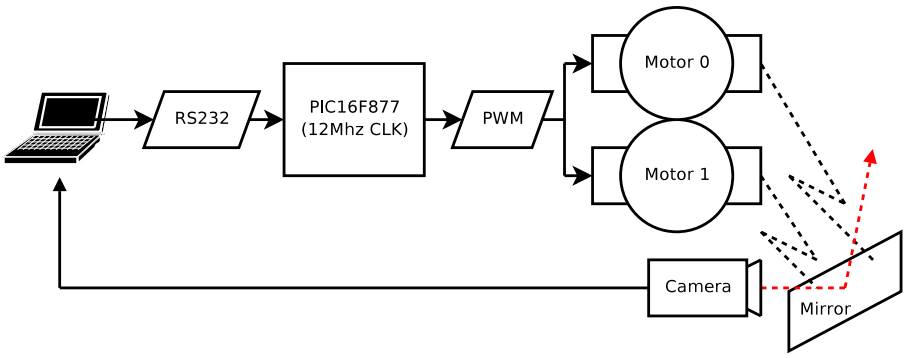


Figure 3.2: Hardware Architecture

3.3 Software Design

The software design is based on driving the control electronics. RS232 is used as the communication standard between the host computer software and control electronics firmware.

Using RS232 to control the servos will require both host software and software for the microcontroller. The role of the microcontroller will be to read received RS232 packets and to control the appropriate servo. This is done via an interrupt driven process. Interrupts are flagged if there is a new RS232 packet or if its time to toggle the motor pin (to generate the PWM). The host PC simply requires software to send RS232 packets of the defined format to a serial port. Ideally a kernel driver should be written to minimise CPU overhead since the software is run in kernel space. However due to time limitation the software developed to control the firmware was programmed to run in user space.

There are two pieces of information we wish to send to the hardware: motor position and which motor. The required bits for the message can be calculated by first calculating the number of positions needed:

$$NumberOfPositions = \frac{FieldOfView}{AngularResolution} \approx 667 \tag{3.3}$$

So 10 bits could represent all the positions available (and allow support for a higher resolution) and since one bit is needed to define which motor will be controlled, the total number of bits needed is 11. However RS232 signals only transmit 8 bit characters so the communications for this vision system will require two RS232 characters. Figure 3.3

Not Used (bits 12-15)	Position (bits 1-11)	Motor (bit 0)

Figure 3.3: RS232 Packet Data Description

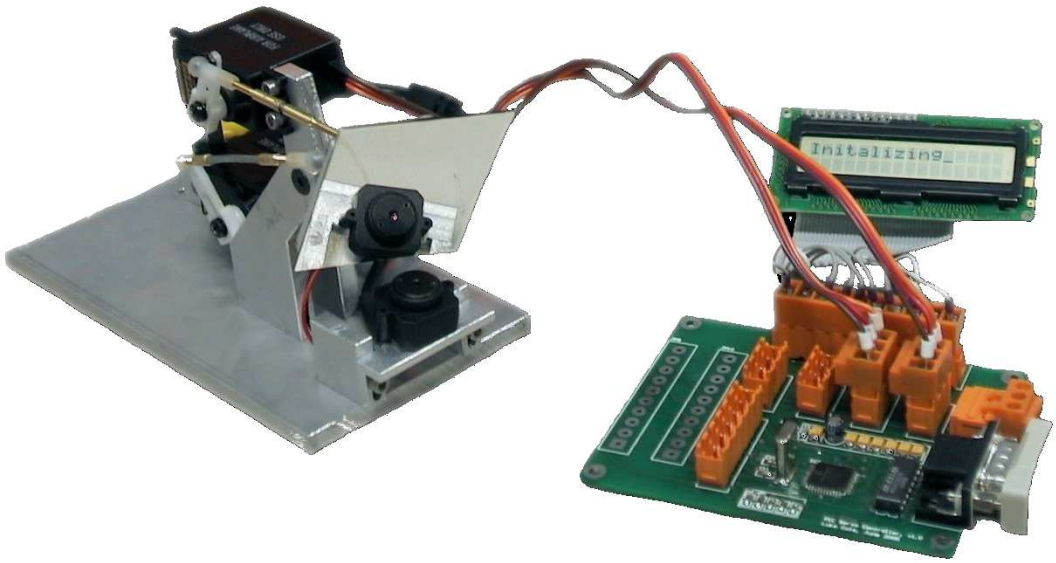


Figure 3.4: Developed Vision System Indicating Primary Components

shows the motor position data layout within a RS232 packet.

There was approximately a total of 2000 lines of code written for the host software and the PIC firmware, approximately 1000 lines each.

3.4 System Overview

Figure 3.4 shows the final developed system indicating components previously mentioned. The final costs for the system can be found in Table 3.5, which is an impressively low cost, compared to most of the other vision system developed in recent years. The system limitations and performance will be discussed in Chapter 6 on page 33.

Item	Qty	Item Cost (AUD)
Digital RC Servo (JR DS8411)	2	150
CMOS Pin-hole camera (Jaycar QC-3454)	1	90
Mirror	1	30
Machining (20 hours @ $40/h)	1	800
Printed Circuit Board	1	100
Electronic Components	1	60
Total Cost		1380

Figure 3.5: Active Mirror Vision System Costs

3.5 Chapter Summary

This Chapter presented the novel active mirror vision system used for the localisation research, and reviewed existing systems and their shortcomings. The hardware design process was then introduced step by step, along with the software design. Finally, the entire system was shown in Figure 3.4 along with a summary of costs in Table 3.5.

Chapter 4

Localisation Software

The localisation software is comprised of a set of algorithms that performs visual Monte Carlo Localisation (MCL). An additional set of algorithms are integrated into the localistion algorithm that introduce a novel technique (view selection) that actively selects, and orients to, the best landmark. This decreases the time for the robot to localise itself and makes the MCL process more efficient. Figure 4.1 shows the integration of the algorithms, where the left of the dotted line represents a visual MCL implementation and the right of the dotted line represents the novel technique, *view selection*, which is a major contribution of this thesis.

4.1 Landmark Map

The landmark map is a list of visual templates within an environment. A single landmark has the following information:

- Location with reference to global coordinates.
- Timestamp when acquired from the camera.
- Angle and distance to camera when acquired.
- Template image of landmark.

This information is stored for the localisation algorithm to use at any point in time. Knowledge of a landmark's global coordinates enables the localiser to estimate the robot's pose from an observation of subset of landmarks. The timestamp is stored to distinguish

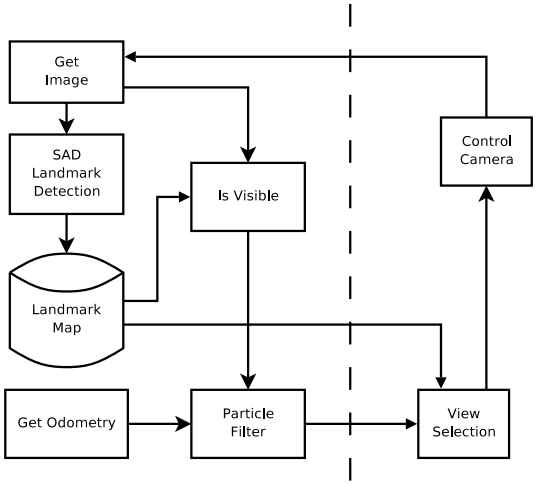


Figure 4.1: Localisation Algorithm

between night and day images, but mainly as a record for testing. The angle and distance to the camera at the time it was taken gives the landmark's approximate orientation and physical size. Storing an image of the landmark is vital in order for recognition or visual correlation.

The map is built by taking different images when scanning the environment and then using the SAD landmark detection algorithm (Watman et al. 2004) to extract landmarks for each image as shown in Figure 4.2. Landmarks are visually distinct parts of the image, that is parts of the scene that are recognisable. Once a reasonable distribution of landmarks is found within the environment, one manually measures each landmark to some global reference along with the angle and distance to the camera and time it was taken. This information along with the landmark image are then manually entered into the map. Figure 4.3 shows the bird's eye view of the environment used and a subset of the landmarks found.

4.2 Known Algorithms

The function of the known algorithms shown on the left side of the dotted line in figure 4.1 is to apply the particle filter, which was discussed in Section 2.6. However, this particle filter has been adapted to perform visual localisation. That is, the importance factors are

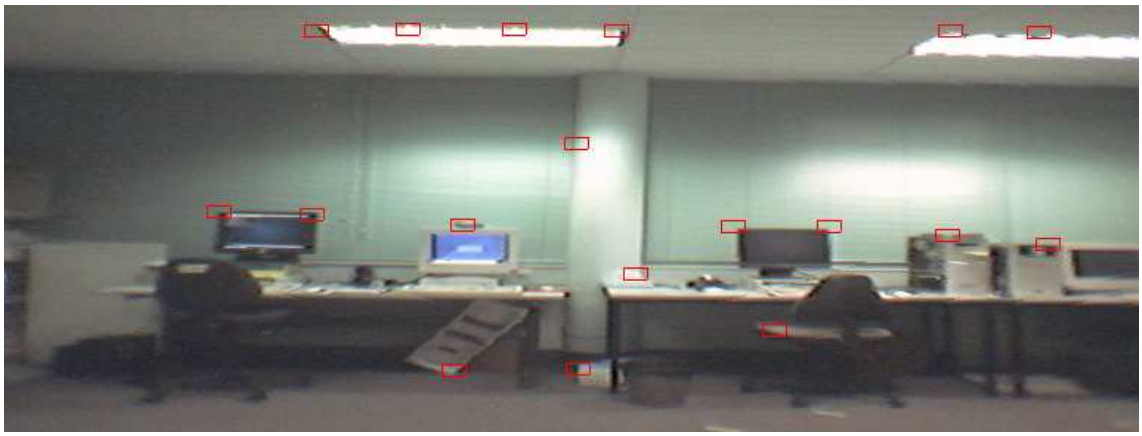


Figure 4.2: Landmarks Extracted via the SAD Landmark Detection Algorithm

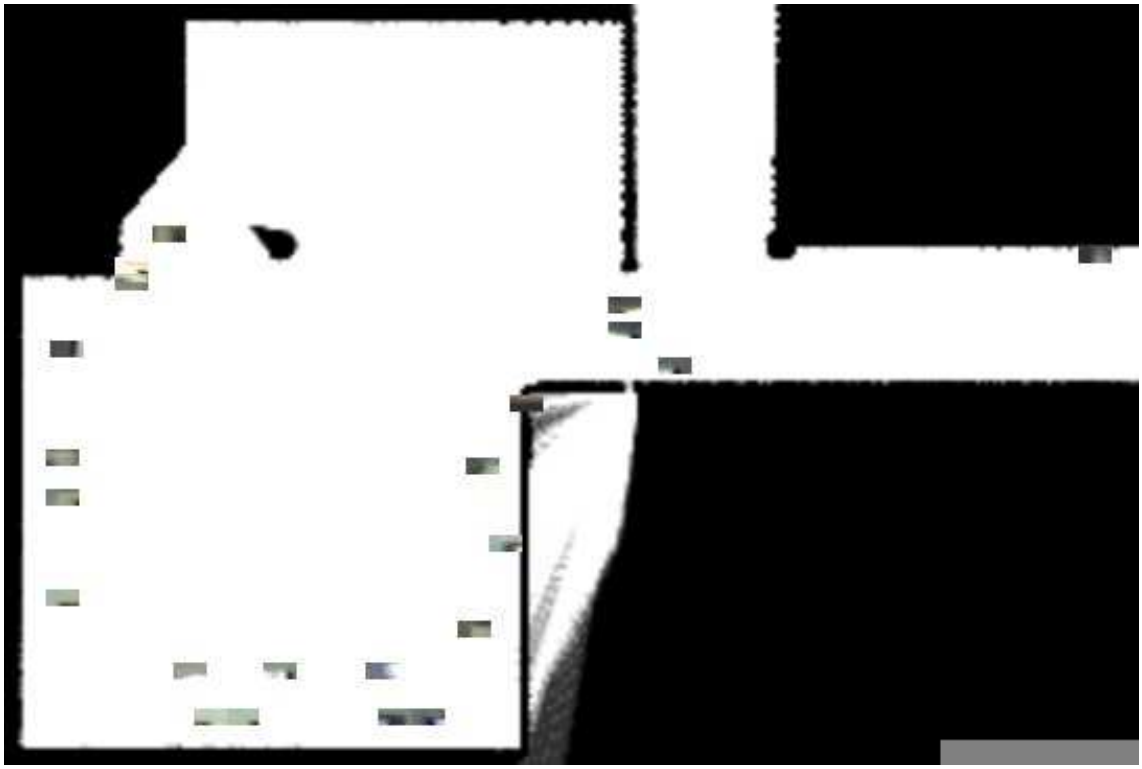


Figure 4.3: Bird's Eye View of Landmark Map

derived from visual information (camera images). More precisely:

$$p^i = 1 - \frac{1}{n\sigma} \sum_{k=0}^{n} s_k \tag{4.1}$$

where p^i is the importance factor for the i^{th} particle, n is the number of landmarks, s_k is the score for the sum of absolute differences (SAD) between the k^{th} landmark and the new image, and σ is a constant defined by:

$$\sigma = TemplateWidth \times TemplateHeight \times BypesPerPixel \times MaxPixelIntensity \tag{4.2}$$

If a landmark is not *visible*, that is, it is not within the camera field of view for a given particle and camera angles, the SAD score for that landmark should be σ. Determining if a landmark is visible or not, is performed by the *IsVisible* algorithm, which simply maps the landmark global coordinates (in millimeters) to the image plane (in pixels), and if the coordinates exceed the image size, the landmark is *not* visible.

4.3 View Selection Algorithm

At any position there are going to be landmarks around the robot that improve the probability distribution of the pose better than others. This thesis has developed an algorithm termed *view selection* that determines a best place to look (landmark), which will make the localisation process more efficient and in turn decrease the time taken for the robot to localise itself. The best landmark to orient to, is determined by weighting each landmark based on:

- Likelihood of observing a landmark.
- Re-orientation time to landmark.
- How much the landmark will reduce the spread (variance) of the probability distribution.

These weightings are then averaged for each landmark and the mirror oriented to the landmark with the highest average weight.

```
function VisWeight(pose, landmark) {

  if (IsBehindWall(pose, landmark))
    return 0.0;

  if (NotFacing(pose.a, landmark.a))
    return 0.0;

  w0 = landmark.depth / Distance(landmark, pose);
  w1 = cos(pose.a, landmark.a);
  weight = (w0 + w1) / 2.0;

  if (weight > 1.0)
    return 1.0;

  return weight;
}
```

Figure 4.4: Algorithm to Calculate the Visibility Weight

4.3.1 Likelihood of observing a landmark

A landmark may not be visible at particular robot poses. So a weight for each landmark is calculated based on the differences between the current depth and orientation to the landmark from the map's landmark depth and orientation information. This weight will be known as the *visibility weight*. The algorithm, as given in figure 4.4, has two considerations apart from calculating a weight, which is to determine if a landmark is behind a wall, and if it is not facing the robot, in either case, the visibility weight is set to zero.

A landmark was defined to be behind a wall by using a grid map of cells defining the valid environment area and if any of the cells in the line of sight from the robot to the landmark are invalid the landmark is said to be behind a wall. Determining the facing orientation is based on the difference between the landmark and the pose orientation. It is assumed that if the orientation difference is greater than 90° the robot is facing the back of the landmark and therefore the landmark weight is set to zero. Figure 4.4 shows the algorithm used to calculate the visibility weight for a landmark given the mean pose (mean of particles).

```
function ReOrienationTimeWeight(pose, landmark) {

  new_camera_angles =
    CalCameraAngles(pose, camera_angles, landmark);

  if (ExceedsVisionSystemLimits(new_camera_angles))
    return 0.0;

  if (new_camera_angles.x == 0 &&
      new_camera_angles.y == 0)
    return 1.0;

  t = ReOrientationTime(camera_angles, new_camera_angles);

  if (t > MAX_ORIENTATION_TIME)
    return 0.0;

  return 1.0 - t / MAX_ORIENTATION_TIME;
}
```

Figure 4.5: Algorithm to Calculate the Orientation Time Weight

4.3.2 Re-orientation time

Weighting the re-orientation time is a rather simple function, however, there are still some considerations such as: the landmark may be currently visible; and, the landmark may be outside the orientation limits. Figure 4.5 shows the algorithm used to calculate the re-orientation time weight for a landmark given the *mean pose*.

4.3.3 Shape of probability distribution

The approximation of the robot's position is presented by the shape of probability distribution, so the smaller the spread of the distribution, the better the estimate of the robot's pose. To reduce the spread, the vision system should look *perpendicular* to the primary axis of the spread. The primary axis is simply the first principal component (eigenvector) of the covariance matrix (Lay 2000) from the particles.

```
function LandmarkProbDistWeight(particles, landmark) {

  covariance_matrix = MatrixCovariance(particles);
  eigenvectors = MatrixEigenVectors(covariance_matrix);
  angle = AngleDiff(eigenvector[0], landmark);

  return ABS(sin(angle));
}
```

Figure 4.6: Algorithm to Calculate the Probability Distribution Weight

4.3.4 Integration

Integrating the above algorithms to obtain a single weight for each landmark could be performed in a number of ways. For simplicity, this thesis just finds the mean weight of the visibility, re-orientation time and probability distribution weights for each landmark. Ideally one could empirically derive scaling of each weight based on performance statistics.

4.4 Chapter Summary

This Chapter presented the visual Monte Carlo Localisation algorithm using the new view selection technique. This involved a discussion on the visual landmark map used and how it was acquired. The visual aspect to the MCL was then outlined. Finally, the primary contribution of this thesis, view selection, was discussed by a look into the various considerations when selecting a landmark to orient to.

Chapter 5

Active Mirror Vision System Characteristics

The developed hardware and control software were tested in a performance evaluation to determine the vision system characteristics. However, the vision system must be calibrated first, that includes, both the camera and the parallel mechanism of the system. Speed and accuracy tests were then performed and evaluated against the performance specifications in Chapter 3. Saccade rate, angular resolution and repeatability were measured to determine the angular velocities, accelerations and accuracy of the vision system. These tests not only demonstrate the performance of the mechanical hardware, but also showed the performance of the control electronics and software. The testing processes will now be discussed along with the results.

5.1 Calibration

Both the camera and the mechanical system need to be calibrated in order to control the vision system correctly and to acquire undistorted images. The camera calibration was performed using the Matlab Calibration Toolbox, which calculated the camera calibration information given in Table 5.1. The toolbox was extended to save a file that maps each pixel to its undistorted true pixel location. This enabled the localistion algorithm to remove the barrel distortion online.

Since the vision system has coupled axes (that is panning the mirror will also cause the mirror to tilt and vis-versa) the mechanical system needs to be calibrated in order to

Measurement	**Pixels $\pm$ Error**
Focal Length	[650 337] $\pm$ [5.21 2.76]
Principal Point	[401 132] $\pm$ [6.85 3.2]
Skew	[0] $\pm$ [0]
Distortion (radial and tangential)	[-0.247 0.039 -0.001 -0.028 0] $\pm$ [0.014 0.027 0.002 0.001 0]
Pixel Error	[1.55 1.617]

Table 5.1: Camera Calibration Results: Matlab Calibration Toolbox Output

calculate the position transformations. The pan and tilt orientation position transformation derived were:

$$\theta_x = \phi_0 \frac{a}{r_0} - atan(\frac{Z_0}{b}) \tag{5.1}$$

$$\theta_y = \phi_0 \frac{a}{r_0} - atan(\frac{Z_0}{b}) \tag{5.2}$$

$$\theta_x = \phi_1 \frac{a'}{r_1} - atan(\frac{Z_1}{b'}) \tag{5.3}$$

$$\theta_y = \phi_1 \frac{a'}{r_1} - atan(\frac{Z_1}{b'}) \tag{5.4}$$

which yields the relationship between the servos angles ϕ_0, ϕ_1:

$$\phi_0 \frac{a}{r_0} - \phi_1 \frac{a'}{r_1} = atan(\frac{Z_0}{b}) - atan(\frac{Z_1}{b'}) \tag{5.5}$$

$$\tag{5.6}$$

where,

$$a = \sqrt{X_0^2 + Y_0^2 + Z_0^2} \tag{5.7}$$

$$b = \sqrt{X_0^2 + Y_0^2} \tag{5.8}$$

$$a' = \sqrt{X_1^2 + Y_1^2 + Z_1^2} \tag{5.9}$$

$$b' = \sqrt{X_1^2 + Y_1^2} \tag{5.10}$$

For these equations θ_x is the pan orientation angle and θ_y the tilt orientation angle of

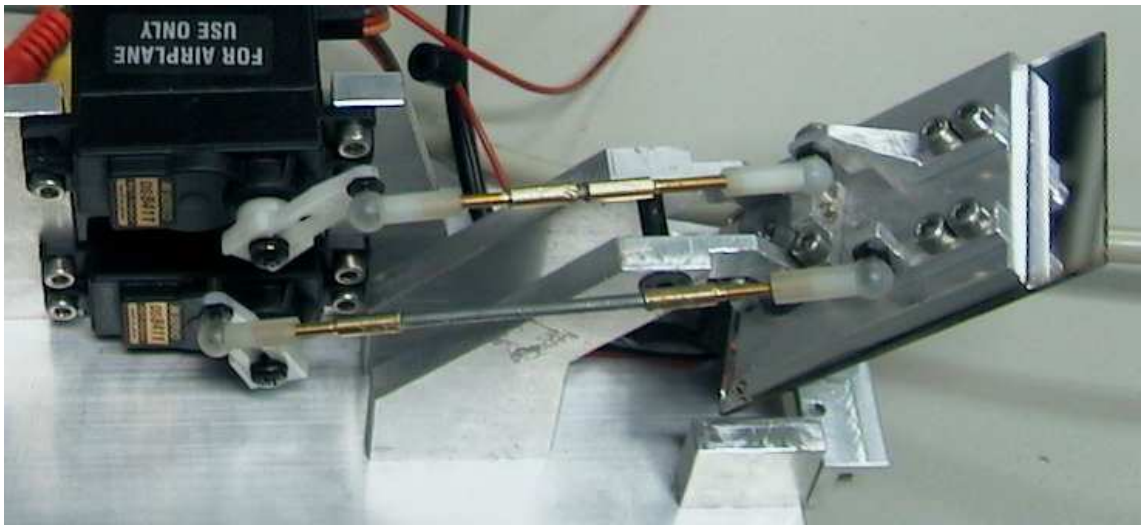


Figure 5.1: Parallel mechanism driving mirror

the vision system, ϕ_i and r_i are the orientation angle and arm length of the i^{th} RC servo. X_i, Y_i are the coordinates of the i^{th} ball joint on the back of the mirror, in reference to the omni joint shown in figure 5.1. The system developed had the parameters:

$$\begin{bmatrix} r_0 \\ r_1 \end{bmatrix} = \begin{bmatrix} 20.0mm \\ 20.0mm \end{bmatrix}, \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} = \begin{bmatrix} 0.0mm \\ 12.0mm \\ 6.0mm \end{bmatrix}, \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = \begin{bmatrix} 20.0mm \\ 20.0mm \\ 0.0mm \end{bmatrix} \tag{5.11}$$

5.2 Speed Test

A simple software program was written to test the speed of the vision system by acquiring time stamped camera images, while driving each of the servos in a cyclic fashion. This means orienting a servo arm back and forth (saccading). The experiment procedure was:

1. A target grid with 5cm spacing was placed on a wall 2m from the vision system (Figure 5.2 shows the experimental setup).

2. The servos were commanded to move backward and forward. Acquiring images at each position recording the current servo positions and time.

3. The recording data determines the angular velocity, angular acceleration and saccade rate.

Another program was written to determine the angular velocity and acceleration by correlating the images. The results can be found in Table 5.2.

5.3 Accuracy

To determine the accuracy of the vision system the *repeatability*, *angular resolution* and *coordinated motion* must be determined. Since the prototype vision system is monocular, coordinated motion of two cameras will not be tested. The experiments used were similar to the speed tests, however the repeatability test had the following difference:

1. Joints were centered.
2. Joints were then commanded to move in random positions a few times.
3. Joints were then centered and the whole process repeated.

The angular resolution can be measured by studying the recorded data from any of the tests, by simply calculating the average angle for a single servo step (distance between two servo positions). The results can be found in Table 5.2.

5.4 Limitations

As with any motion system, there are going to be limits, using a mirror means the camera may capture the camera, or vision system itself as shown by Figure 5.2. Here the black region is the camera.

5.5 Evaluation

The overall performance of the prototype shows real potential for the mirror vision system idea. The key problem of the system was the microcontroller selected to control the servos. It was assumed a 5Mhz interrupt could be generated given that the instruction cycle was 200ns, however the best interrupt that could be generated was only 200kHz.

Figure 5.2: Left: Experimental Setup, Right: Limitations of Mirror Vision System

Figure 5.2 shows the characteristics of the mirror vision system along with a comparison against the design requirements.

Specification	Unit	Design Value	Measured Tilt	Measured Pan
Saccade Rate	Hz	5	3Hz	5Hz
Angular Resolution	$^\circ$	0.12	0.4	0.4
Angular Repeatability	$^\circ$	0.01	0.1	0.1
Maximum Range	$^\circ$	60	45	22.5
Maximum Velocity	$^\circ.s^{-1}$	600	666	666
Maximum Acceleration	$^\circ.s^{-2}$	600	666	666

Table 5.2: Characteristics of Mirror Vision System

5.6 Chapter Summary

This Chapter has shown that the design requirements were largely met and therefore an effective vision system was developed on a hardware and software level. The tests prove that the system would be a good vision system for localisation since it matches the human-like eye movements and is low in cost.

Chapter 6

Localisation Experiments and Results

The localisation experiments were designed to show the effectiveness of the developed localisation software with and without the mirror vision system. Chapter 4 showed that the localisation software is comprised of two Sections, one of these being *view selection*, which can operate independently from the core of the Localiser (AMCL). An experiment was carried out to test the localisation software without the new view selection technique (and hence not orienting the mirror vision system). This was followed by experiments to test components of view selection and then a final experiment was carried out to test the entire localisation software developed. These experiments show how the view selection technique significantly improves localisation software.

6.1 Using No View Selection (No Camera Movement)

This experiment tests the effectiveness of the localisation, which will enable comparison against current visual localisers and give the view selection technique a benchmark to improve on. The process was:

1. Place the robot at a known location so it is surrounded by landmarks from the landmark map and centre the servos.

2. Start the visual localisation software without view selection.

3. Control the robot to move to a new known location. And record the visual particle filtering process given by the localiser.

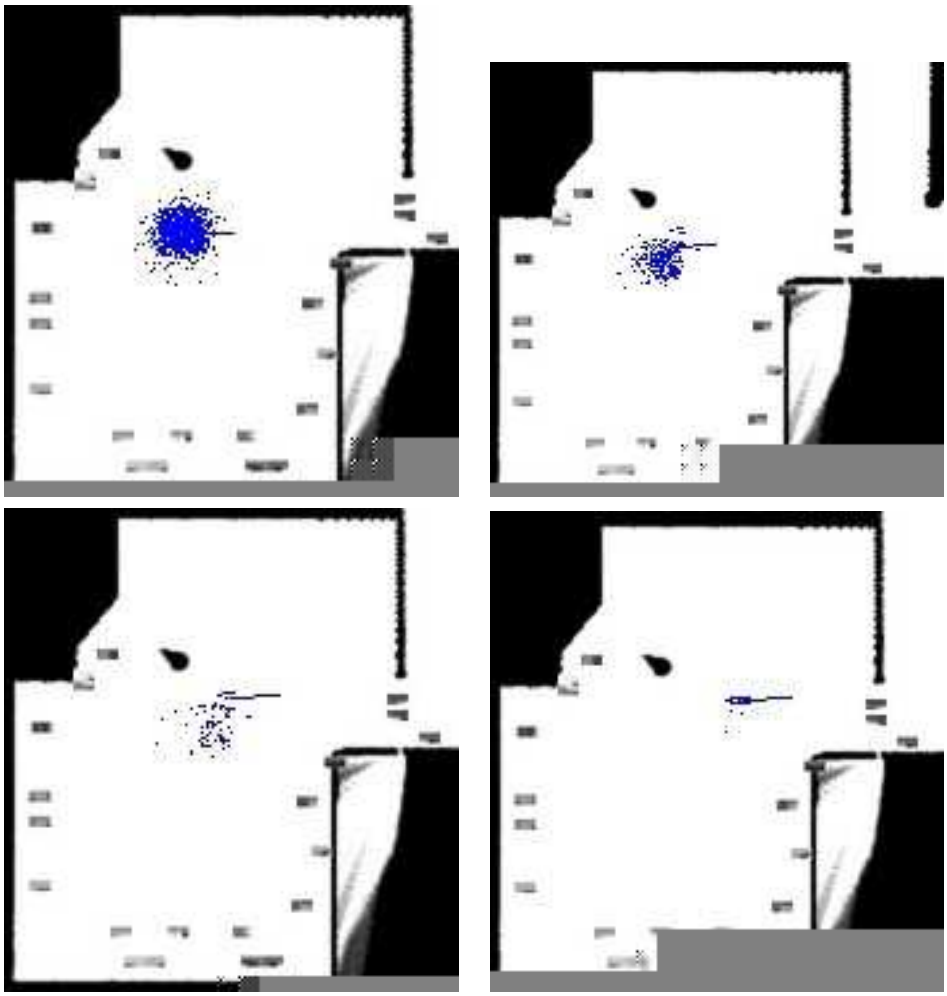


Figure 6.1: Localisation Process Output Without View Selection (One Frame Per Second)

4. Stop the robot and measure its pose, compare this to the localiser pose.

5. Repeat for different frame rates.

Figure 6.1 shows an image series of the particle filter process displayed by the localiser. The small blue dots are the particles, which is the probability distribution of the robot's pose, and as can be seen the particle spread reduces as the robot moves around and acquires observations. The large blue arrow is the mean pose of the approximation. Figure 6.3 shows a graph of the of error after travelling 3m for various frame rates, which shows the vision system can best localise itself when the frame rate is at 1Hz.

6.2 Using View Selection Technique

Four experiments were carried out in order to test the view selection system, both the components independently and the integrated system. This involved a similar process to the experiment run without view selection, however with the different components of view selection enabled and disabled. Figure 6.2 shows the outputs using the various components and when integrated. The top image is considering the visibility of a landmark, the image below that is considering the re-orientation time to a landmark, and the third image is considering the probability distribution of the robots uncertain. The final image shows the output when all three components are integrated. The output of the localiser has some additions to the output discussed in Section 6.1. The large red arrow indicates the camera pan angle the vision system should orient too. The small red arrow indicates the orientation possible within the vision system limits. The three shades of green lines, from the mean pose to landmarks, represent the best landmark for each of the view selection considerations. The darkest green representing the best visibility and the lightest green representing the best landmark that reduce the variance of the probability distribution. Figure 6.3 shows the results of the time to localise versus the frame rate, which shows an major improvement when compared to not using view selection.

6.3 Chapter Summary

This Chapter presented the experiments devised to test the effectiveness of the developed visual localiser. This included testing the localisation software without view selection, which means the active component of the active mirror vision system is disabled. The active component was then tested in four experiments, three of these using components of view selection and the final experiment using the integratation of all the view selection components. The results show that view selection is an extremely promising technique that reduces the time to localise and adds efficiency for localisation based on a particle filter implementation.

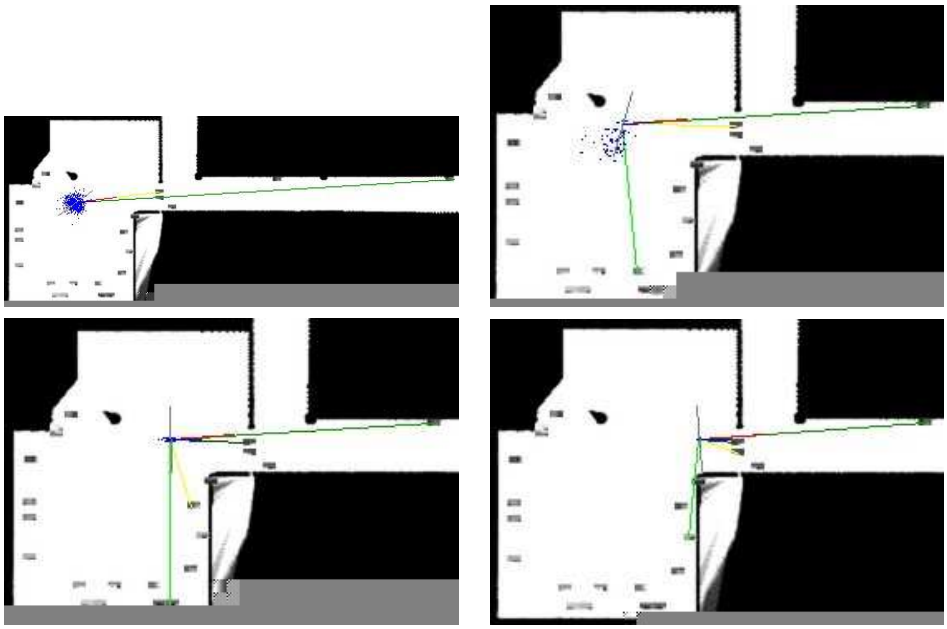


Figure 6.2: Localisation Process Outputs Using View Selection (One Frame Per Second)

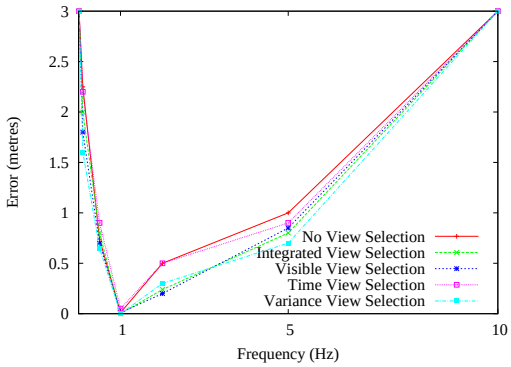


Figure 6.3: Localisation Results (Traveled 3m)

Chapter 7

Conclusion

Using the active mirror vision system and view selection have proven to be suitable techniques for robot localisation. The mirror based active vision system developed shows real potential as a solution to active vision. The view selection technique worked as anticipated, both in simulation and, using a real robot in a real world, in real time.

The cost of the vision system was remarkably low compared to existing active vision systems. Considering the performance of these systems are not much different from the active mirror vision system, one can see the advantages and potential of using a mirror. While the parallel mechanism causes the axes to be coupled, which can be removed by software, such a mechanism provides mechanical strength and has a low moving mass. Furthermore, less maintenance is required due to the choice of transmission, which makes this active mirror vision system a practical solution. The selection of components provided the desired response, however the microcontroller selected was found to limit the angular resolution. But this is no major concern since its just a matter of changing the microcontroller. One issue with a mirror based vision system is the mirror must remain clean, however the real world may cause it to get scratched, in which case, a cover could be developed to ensure the system is more reliable on a long term basis.

View selection has presented some interesting results, and it could even be said that the active vision system mimics human (or animal) eye expressions, since it rapidly scans the environments. Though its not truly scanning the environment, its *view selecting* the best landmarks. In any case, the components of view selection, both independently and integrated, have improved the localisation process extremely well. The *IsVisable* algorithm developed, to define the importance factors for visual localisation, worked reason-

ably well. The major problem with the *IsVisable* algorithm is that the localisation time is proportional to the size of the search windows within the image. The visual localisation algorithm developed not only proved itself to be a solution to localisation but adds a sense of awareness in the way it rapidly investigates its environment, similar to a human lost in a travelling circus.

7.1 Review

This thesis investigated the problem of high-speed vision-based localisation for mobile robots. By the study of two new techniques: a active mirror vision system and view selection for localisation.

The study showed that an active mirror vision system is a promising system for active vision due to its cost, size, performance and reliability. The study also showed that the view selection technique allows the localisation process to be efficient in time and computer computation. So it provides a powerful contribution to the localisation process.

Even though the project was a success, the limitation on time stopped further development (see Section 7.2) of ideas for localisation using an active mirror vision system. Though the results in this thesis show enough evidence to prove these techniques are a promising solution to robot localisation.

7.2 Discussion and Future Work

The results of the localisation algorithm developed follow the theory desired in Chapter 2. The performance of the active vision system was impressive, however the resultant performance was less than what was derived, simply due to the microcontroller used.

Using SAD to correlate landmarks also presents some advantages and disadvantages to the visual localiser. Using SAD was a good choice because it captures all the information present. However, correlation is computationally expensive. Some other feature based detection algorithm may provide a faster solution,however this using correlation has illustrated that view selection is useful, no matter what type of matching technique is used.

Making the software invariant to lighting conditions would be the next step for future work on this localiser. Lighting conditions limit the performance and different times of

the day. A method for this would be to drive the robot around at different times of the day, on different days. Then determine the landmarks that are most stable, which could be used for the landmark map.

There are a few modification, that can be made to the active mirror vision system. The only problem with the control electronics/software was the microcontroller. It needs to be upgraded to support a higher interrupt rate for the desired resolution. Though there are a few mechanical changes that could be employed into the vision system, as with any prototype. The major change would be to machine more of mirror bracket so the omni joint can orient further and hence increase the pan viewing angle. The fact the system is coupled is also something which would be desirable to remove.

Future work for active mirror vision:

- Explore different materials such as plastic.
- Upgrade communication standard to USB2.0 for future compatibilities.

Future work for visual localisation and view selection:

- Explore different methods to deriving the importance factor of possible poses.
- Explore other methods to reduce the search window size of visible landmarks.
- Automatic selection of stable landmarks.
- Integrating into simultaneous localisation and mapping (SLAM).

Appendix A

Camera Datasheet

MINIATURE SURVEILLANCE COLOR CAMERA

MODEL: 50-50

THANK YOU FOR BUYING OUR PRODUCT. PLEASE READ AND FOLLOW THESE INSTRUCTIONS CAREFULLY.

Contents

Camera with three leads.
Battery connection clip.
AC/DC adapter

About your camera
This camera simply plugs into the standard television or video recorder via the three leads and A/V cables (not supplied).

The camera has a mini microphone to offer sound as well as sight. It can be used to monitor callers at your door, to keep an eye on your car, to check on sleeping babies and children playing; or to keep an eye on any other area of your home or any area of your office, factory or shop.

Camera specification:
1. Power: DC 9V±1V, 100MA.
2. TV system: PAL.
3. Video output: 1Vp-p 75ohm.
4. Resolution: 380 TV lines.
5. Lux: 4 Lux.
6. Lens: f=5.2mm, Viewing angle=60°.
7. Dimension: 20 x 20 x 16mm.

Connection
Before you fix your camera in the final location, it is advisable to ensure it is working to your satisfaction first. The black lead from the camera is for power in (DC 9V±1V), you can either use the adapter supplied or use a 9V DC battery via the battery connection clip supplied with the camera.

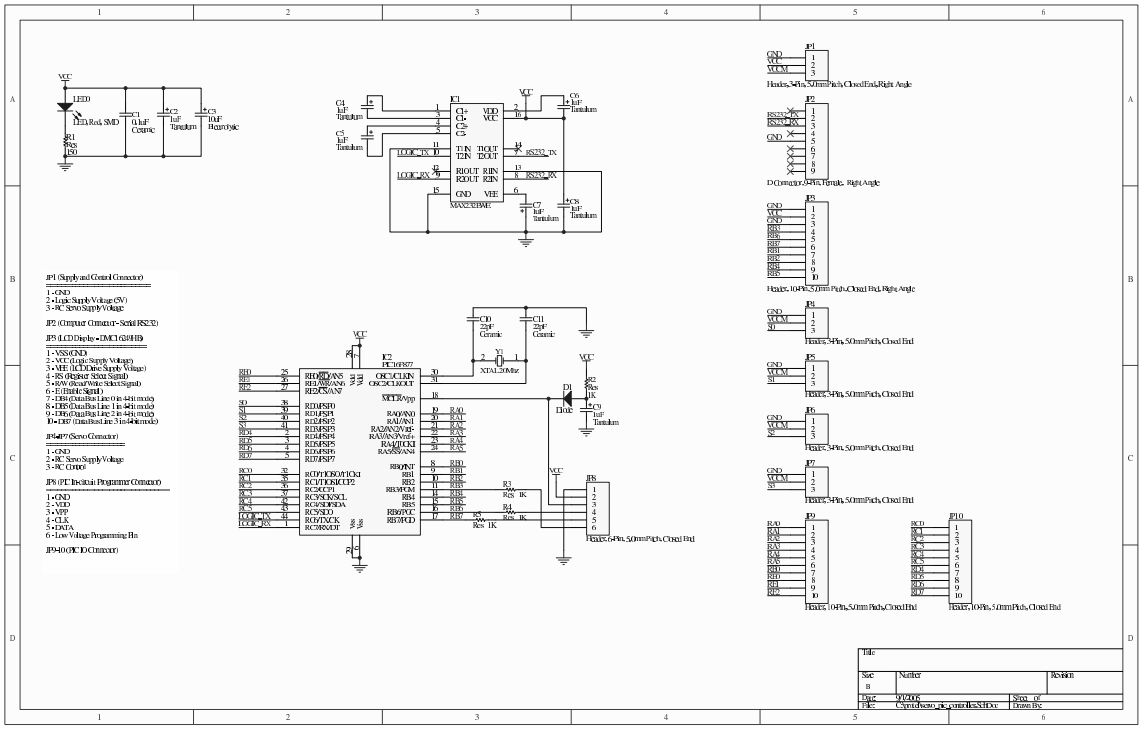
Important note: If use adapter, please only use the adapter supplied with this product. Other adapters may not work with the camera and may damage the camera.

Appendix B

Servo and Ball-Joint Datasheets

Appendix C

Electronic Schematic



Bibliography

Borenstein, J. (1995), 'Control and kinematic design of multi-degree-of-freedom robots with compliant linkage', *IEEE Transactions on Robotics and Automation* .

Brooks, R., Breazeal, C., Marjanovic, M., Scassellati, B. & Williamson, M. (1998), 'The cog project: Building a humanoid robot', *Springer Lecture Notes in Artificial Intelligence* **1562**.

Cassinis, R., Tampalini, F. & Fedrigotti, R. (2005), Active markers for outdoor and indoor robot localization, *in* 'R.T.2005-02-43'.

Chang, C. & Gaudiano, P. (2000), 'Biomimetic robotics', *Robotics and Autonomouse Systems* **30**.

Clement, M. (1996), 'On the development of the agile eye', *IEEE Robotics and Automation Magazine* pp. 29–37.

Couch, L. (2001), *Digital and Analog Communication Systems*, Prentice Hall.

Cox, I. (1991), 'Blanche - an experiment in guidance and navigation of an autonomous robot vehicle', *IEEE Transactions on Robotics and Automation* **7**(2), 193–204.

Cox, I. & Wilfong, G. (1990), *Autonomous Robot Vehicles*, Springer-Verlag.

EPFL Stereo Vision Sphere (1998). http://diwww.epfl.ch/lami/team/carmona/epfl_stereo_platform.html.

EPFL Vision Sphere (1997). http://diwww.epfl.ch/lami/team/carmona/project-seb.html.

Fleischer, J. & Troxell, W. (1999), 'Bio-mimcry as a tool in the design of robotic systems', *Proceedings of the 3rd International Conference on Engineering Design and Automation* .

Fox, D., Burgard, W. & Thrun, S. (1999), 'Markov localization for mobile robots in dynamic environments', *Journal of Artificial Intelligence Research* **11**.

Fox, D., Thrun, S., Burgard, W. & Dellaert, F. (2000), *Sequential Monte Carlo Methods in Practice*, Forthcoming.

Franz, M. & Mallot, H. (2000), 'Biomimetic robot navigation', *Robotics and Autonomouse Systems* **30**.

GTK (2005). http://www.gtk.org/.

Harmonic Drive Systems (2005). http://www.hdsi.net.

Heuring, J. & Murray, D. (1999), 'Modeling and copying human head movements', *IEEE transactions on Robotics and Automation* **15**(6), 1095–1108.

Jensfelt, P. (2001), *Approaches to Mobile Robot Localization in Indoor Environments*, Royal Institute of Technology, Stockholm, Sweden.

Kandel, E., Schwartz, J. & Jessell, T. (2000), *Principles of Neural Science, 4th edition*, Appleton and Lange.

Kuniyoshi, Y., Kita, N., Rougeaux, S. & Suehiro, T. (1995), 'Active stereo vision system with foveated wide angel lenses', *Asian Conference on Computer Vision* .

Lay, D. (2000), *Linear Algebra and its Applications 2nd edition*.

Leonard, J. & Durrant-Whyte, H. (1991), 'Mobile robot localization by tracking geometric beacons', *IEEE Transactions on Robotics and Automation* .

Leonard, J. & Durrant-Whyte, H. (1992), *Directed sonar sensing for mobile robot navigation*, Kluwer, Dordrecht, The Netherlands.

Madden, B. & von Seelen, U. C. (1995), 'Penneyes - a binocular active vision system'. Technical Report MS-CIS-95-37/GRASP LAB 396.

Mifflin, H. (1995), *The American Heritage Stedman's Medical Dictionary*, Houghton Mifflin Company.

Mircochip (2001), 'pic16f877'. http://ww1.microchip.com/downloads/en/DeviceDoc/30292c.pdf.

Moravec, H. (1998), 'When will the computer hardware match the human brain?', *Journal of Evolution and Technology* **1**.

Pahlavan, K. & Eklundh, J. (1992), 'A head-eye system - analysis and design', *CVGIP: Image Understanding: Special issue on purposive, qualitative and active vision* .

Panzieri, S., Pascucci, F., Setola, R. & Ulivi, G. (2001), A low cost vision based localization system for mobile robots, *in* 'Proceendings of the IEEE Mediterranean Conferences'.

Reminder on Parallel Kinematics (2003). http://lsro.epfl.ch/prd/definitions.php.

Rofer, T. & Jungel, M. (2004), Fast and robust edge-based localization in the sony four-legged robot league, *in* '7th International Symposium on RoboCup 2003 (Robot World Cup Soccer Games and Conferences)'.

Roumeliotis, S. (1999), 'Reliable mobile robot localization'.

RTK Library (deprecated) (2004). http://umn.dl.sourceforge.net/sourceforge/playerstage/librtk-src-2.3.0.tar.gz.

Salichs, M. A. (2001), 'Navigation of mobile robots: Learning from human beings', *Plenary Session. IFAC Workshop on Mobile Robot Tecnology. Jejudo Island. Korea* .

Scassellati, B. (1998), 'A binocular, foveated active vision system'. MIT AT Memo 1628.

Sharkey, P., Murray, D., Vandevelde, S., Reid, I. & McLauchlan, P. (1993), 'A modular head/eye platform for real-time reactive vision', *Mechatronics Journal* **3**(4), 517–535.

Shoval, S. & Borenstein, J. (1995), 'Measurement of angular position of a mobile robot using ultrasonic sensors'.

Singhal, A. (1997), 'Issues in autonomous mobile robot navigation'. A survey paper towards partial fulfillment of MS degree requirements.

Thompson, S., Matsui, T. & Zelinsky, A. (2000), 'Localisation using automatically selected landmarks from panoramic images', *Australian Conference on Robotics and Automation* .

Thrun, S. (1998), 'Bayesian landmark learning for mobile robot localization', *Machine Learning* **33**(1), 41–76.

Thrun, S., Fox, D., Burgard, W. & Dellaert, F. (2000), 'Robust monte carlo localization for mobile robots', *Artificial Intelligence* **128**(1-2), 99–141.

Truong, H. (2000), *Active Vision Head*, ANU Publishing.

Tsujita, T., Konno, A. & Uchiyama, M. (2005), 'Design and development of a high speed binocular camera head', *IEEE International Conference on Robotics and Automation* .

Watman, C., Austin, D., Barnes, N., Overett, G. & Thompson, S. (2004), 'Fast sum of absolute differences visual landmark detector', *International Conference on Robotics and Automation* **5**, 4827–4832.

Worsley, F. (1978), *Shackleton's boat journey*, New York.