Chapter from the book
Downloaded from: http://www.intechopen.com/books/

Interested in publishing with IntechOpen?
Contact us at book.department@intechopen.com
A Behaviour-Based Architecture for Mapless Navigation Using Vision

Regular Paper

Mehmet Serdar Guzel¹,* and Robert Bicker²

School of Mechanical and System Engineering, Newcastle University, UK
* Corresponding author E-mail: m.s.guzel@newcastle.ac.uk

Received 11 Sep 2011; Accepted Jan 05 2012

DOI: 10.5772/46200

Abstract Autonomous robots operating in an unknown and uncertain environment must be able to cope with dynamic changes to that environment. For a mobile robot in a cluttered environment to navigate successfully to a goal while avoiding obstacles is a challenging problem. This paper presents a new behaviour-based architecture design for mapless navigation. The architecture is composed of several modules and each module generates behaviours. A novel method, inspired from a visual homing strategy, is adapted to a monocular vision-based system to overcome goal-based navigation problems. A neural network-based obstacle avoidance strategy is designed using a 2-D scanning laser. To evaluate the performance of the proposed architecture, the system has been tested using Microsoft Robotics Studio (MRS), which is a very powerful 3D simulation environment. In addition, real experiments to guide a Pioneer 3-DX mobile robot, equipped with a pan-tilt-zoom camera in a cluttered environment are presented. The analysis of the results allows us to validate the proposed behaviour-based navigation strategy.

Keywords Mapless Navigation, Scale Invariant Feature Transform, Neural Network, Visual Homing, Subsumption Architecture.

1. Introduction

A number of potential markets are slowly emerging for mobile robotic systems. Entertainment applications and household or office assistants are the primary targets in this area of development. These types of robots are designed to move around within an often highly unstructured and unpredictable environment. Existing and future applications for these types of autonomous systems have one key problem in common, navigation [1]. One of the most popular approaches used to handle with such a complex system is to decompose the global task into several simpler, better-defined behaviours [2]. Behaviour-based approaches have become dominant methodologies for designing control schemes for robot navigation. A behaviour-based system has multiple integrated competences in which each level of competence is implemented by layers in the control system. Individual layers can work for individual goals simultaneously [3]. Each individual layer generates behaviours that provide interaction between the robot and its environment, based on gathered data from different sensors. Vision is one of the most powerful and popular sensing methods used for autonomous navigation. Compared with other on-board sensing...
techniques, vision-based approaches to navigation continue to demand a lot of attention from the mobile robot research community, which is largely due to their ability to provide detailed information about the environment that may not be available using combinations of other types of sensors. Integration of a powerful machine vision system to behaviour-based architecture is one of the most important aspects of such types of system. Even though the past decade has seen the rapid development of vision-based sensing for indoor navigation tasks, vision-based navigation for mobile robots is still an open area of research. Vision-based navigation falls into three main categories depending on the localization methods [4]. The first category is called map-based navigation, which requires the robot to be provided with a model of the environment. These models may contain different degrees of detail, varying from a complete CAD model of the environment to a simple graph of interconnections or interrelationships between the elements in the environment. The second category is called map building-based navigation. Here a 2-D or a 3-D model of the environment is initially constructed by the robot using its on-board sensors, after which the model is used for navigation. The final category is called mapless navigation where the robot is able to navigate without any prior description of the environment. The required robot motions are determined by observing and extracting relevant information from the landmarks in the environment, such as walls, desks, doorways and relevant equipment. When comparing with other methods, it can be claimed that the mapless strategy resembles human behaviours more than others. The proposed architecture is constructed based on a mapless navigation strategy which is suitable for partially unstructured and unpredictable environments. A precise navigation solution for partially unstructured and unpredictable environments requires a robust and flexible vision strategy. For real-time systems, system performance is another important issue that must be considered and the time required for image processing should be tolerated by the system. Many methods have been proposed for goal-based tasks, based on artificial landmarks or template matching techniques, providing precise but limited solutions to this problem [3, 5, 6, 7, 8 and 9]. As opposed to those methods, in this work a novel method based on monocular vision, involving scale invariant feature transform (SIFT) [10,11] and a variant of the snapshot model of visual homing techniques [7,12] is proposed. The vision system is powered by a camera having pan-tilt-zoom (PTZ) capabilities, which improves the field of view of the system. A snapshot image is taken by a robot at the goal position. The disparity between current and snapshot images is subsequently used to guide the robot’s return. SIFT is used to extract robust features between the snapshot image and the current image. According to the matched feature, the heading angle and velocity of the mobile platform are obtained by a simple but efficient control law. A conventional SIFT algorithm is not suitable for real-time applications, thus the performance of the algorithm is amended by Open Multi-Processing (OpenMP). OpenMP is a free application programming interface (API) that supports multi-platform shared memory multiprocessing programming in C, C++ and Fortran. The SIFT technique has been used previously for vision-based mobile robot localization and mapping algorithms that use scale-invariant image features, such as landmarks, in dynamic environments. However, this paper proposes a map building-based approach and relies on stereo vision [13]. Neural networks, as used in AI, have traditionally been viewed as simplified models of neural processing in the brain, even though the relation between this model and the brain’s biological architecture is debated. A multilayered neural network-based solution is presented to overcome the obstacle avoidance problem. Obstacle avoidance behaviours are successfully implemented using this method. Recently, some studies have been conducted in order to overcome vision mapless navigation problems based on monocular vision. One of those addresses the problem of vision-based navigation and proposes an original control law to perform the navigation which, however, requires precise camera calibration and cannot overcome the obstacle avoidance problem [8]. Another similar study has built a fully autonomous mobile robot comprising several behaviours and employing a vision-based landmark recognition system for robot navigation. This, however, works with artificial landmarks [3]. In addition, some methods requiring a human-guided learning step have been proposed to overcome the mapless navigation problem [9]. This paper is organized as follows. In Section 2, the background knowledge of this study is briefly introduced. The design of the behaviour-based mobile robot system is presented in Section 3, including a neural network-based obstacle avoidance strategy, a visual homing-based navigation system and behaviours. Section 4 provides the implementation of the behaviour-based robot and the experiment results from both the real and simulation experiments. The study is concluded in Section 5.

2. Background

In this section, a snapshot model of visual homing, SIFT and a multilayer artificial neural network will first be outlined, followed by a brief introduction of the behaviour-based approaches.

2.1 Visual homing

Insects are able to return to important places in their environment by storing an image of the surroundings whilst at the goal location and later computing a home location.
direction from a matching between this ‘snapshot’ image and the currently perceived image, which is called visual homing. Various models have appeared in the biological literature, which provide an algorithmic-level description of how insects might use visual homing strategies to return to a goal position. Very similar ideas are pursued for the visual navigation of mobile robots. Visual homing is a type of visual servoing, requiring no explicit quantitative localization and thus no metric map. A robot employing a visual homing algorithm captures an image IS, called the snapshot, at the goal location S = (xS, yS) and, when later attempting to return to this location from a nearby position IC = (xC, yC), compares the current image IC with the snapshot and infers the direction and/or distance to the goal location from the disparity between the two [12]. During this study, the visual homing principle is adapted to a behavioural-based architecture using monocular vision as the primary sensor in order to provide navigation in a partially cluttered environment.

2.1 Scale invariant feature transform (SIFT)

Scale invariant feature transform (SIFT) is an intensity-based feature description algorithm which depends on intensity patterns to find points or regions that satisfy some uniqueness and stability criteria [11]. The algorithm, in which applications include object detection, robot navigation 3-D modelling, video/image tracking and gesture recognition, was proposed by Lowe [10]. Any object in an image is able to provide several features which are interesting points on the object that can be extracted to provide a feature description of the object. This description, extracted from a training image, can then be employed to identify the object when attempting to locate the object in a test image containing many other objects. It is important that the set of features extracted from the training image is robust to changes in image scale, noise, illumination and local geometric distortion, for performing reliable recognition. Lowe’s [10] patented method can robustly identify objects even among clutter and under partial occlusion because the most notable improvements provided by SIFT are invariance to image scaling, rotation and partially invariant to change in illumination and 3-D camera viewpoint. Features are well localized in both the spatial and frequency domains which reduce probability of disruption by occlusion, clutter or noise. In addition, the features are extremely distinctive, which allows a single feature to be accurately matched with high probability against a large database of features, a basis for many applications in computer vision and image processing [10]. The evaluations carried out suggest that SIFT-based descriptors, which are region-based, are the strongest and most distinctive [11], and are therefore particularly suited for feature matching and object detection. However, the main drawback of the algorithm is its computational complexity which usually discourages its real-time utilization. There are four major stages of computation used to generate the set of image features, namely: scale-space extreme detection, keypoint localization, orientation assignment and keypoint descriptor. A brief introduction and an overview of the algorithm involving these major steps are illustrated in Figure 1. For this work, the SIFT algorithm is utilized to find correspondence between the goal and reference images to generate control parameters for the robot.

2.2 Multilayer neural networks

Multilayer networks solve the classification problem for non-linear sets by employing hidden layers, whose neurons are not directly connected to the output. The additional hidden layers can be interpreted geometrically as additional hyperplanes, which enhance the separation capacity of the network. Supervised learning is the training methodology for this type of neural network architecture and the basic idea is to present the input vector to the network. Calculations are done in the forward direction towards the output of each layer and the final output of the network. The desired values for the output layer are known, therefore, the weights can be adjusted as for a single layer network. Instead, a back propagation (BP) algorithm, based on the gradient descent rule, calculates the weight changes in the hidden layers. The error in the output layer is back propagated to these layers according to the connecting weights. This process is repeated for each sample in the training set. One cycle through the training set is called an epoch. The number of epochs required to train the network depends on various parameters, especially on the error calculated in the output layer [14].

![Figure 1. Flowchart of SIFT Algorithm](image-url)
2.3 Behavioural-based robotics

Classic AI approaches to robotics were limited by their computational complexity. To overcome this weakness, a new concept has been proposed which principally defines the idea that intelligence has to be studied in terms of a robot interacting with its environment. According to this approach, it is beneficial to decompose the navigation task into multiple independent task-achieving modules. These modules are called motor schemas and each of them is responsible for a subtask of the whole action. A behaviour module couples motor commands very tightly to sensory readings, has a minimal internal state and solves tasks of limited complexity in a reactive manner. A milestone in the control of behaviour-based robotics was subsumption architecture, which is a way of decomposing complicated intelligent behaviour into many basic behaviour modules, which are in turn organized into layers. Each layer implements a particular goal of the robot and higher layers are increasingly abstract [2]. Each layer’s goal subsumes that of the underlying layers. The priority of the lower levels is higher than the upper levels.

3. Design of a Behaviour-Based System

In this section, two novel approaches will be introduced, the first of which is a robust and simple navigation strategy, inspired from visual homing and visual servoing techniques. The second is a neural network-based solution which will be discussed to address the obstacle avoidance problem.

3.1 Navigation via SIFT based on visual homing

Vision potentially has the most powerful sensing capability to provide reliable and safe navigation. For indoor navigation, researchers rely on artificial landmarks, such as coloured or geometrical objects, to achieve safe navigation. Many approaches which employ artificial landmarks are both easy to design and implement. However, the main disadvantage of these approaches is their dependence on specific tasks. Visual homing provides a good alternative to these methods, but most visual homing techniques rely on omnidirectional vision which captures images in low resolution. Alternatively, stereo vision-based techniques which acquire robust in-depth information, which may not be obtained by monocular vision precisely, are very common. However, these techniques suffer from some serious disadvantages, involving the computational cost of the stereo systems and synchronization problems between the cameras and their calibration. In this section a new alternative method inspired by the visual homing technique relying on the SIFT algorithm will be introduced. The main idea is to generate control variables involving linear velocity and angular velocity (turning rate), based on matching results between the current image and the goal image. Preliminary test results verify that the proposed algorithm estimates turning rate and linear velocity of the mobile vehicle with reasonable accuracy and affordable computational time. The flowchart of the proposed algorithm is illustrated in Figure 2, which will be discussed in detail in the following paragraph. The first part of the algorithm is to enhance the input image against any possible illumination or noise and is called pre-processing. The input image is convolved with a filter based on the first derivative of a Gaussian to obtain a blurred version of the image, which removes unexpected noises and smoothes images. Subsequently, histogram equalization is applied to the filtered image to adjust its contrast. The second function of the algorithm is to extract key features from enhanced images using a SIFT algorithm. As has been previously mentioned, SIFT is one of the most powerful and popular feature detection algorithms, however, due to its computational cost, it is not suitable for real-time applications. In order to cope with this problem, a cross-platform library that computes fast and accurate SIFT image features, which is optimized with SIMD instructions and OpenMP [15], is used instead of the conventional SIFT implementation. The performance of this enhanced algorithm is quite impressive. To validate the performance of this enhanced algorithm with different image resolutions, several experiments were performed, which will be discussed in Section 4.

Figure 2. Flowchart of the SIFT-based control algorithm
The following step is to design an appropriate control law in order to provide the navigation towards one or multiple targets. The proposed algorithm basically utilizes location and magnitudes of extracted features to steer the robot. According to the algorithm, each matched point at the current image is assigned to corresponding location clusters, obtained by dividing the image vertically, and each of these clusters is considered with the total number of matched features. The number of clusters directly depends on the resolution of the acquired images. The proposed architecture works with low resolution images (176x144) and (352x288), hence the estimated cluster size for this problem is four, namely: Left (l), More-Left (lsl), Right (r) and More-Right (rsr) respectively. Distribution (d) of the matched values is a key aspect in order to estimate the next possible turning rate which can be illustrated by the following expressions:

\[
d = \begin{cases} 
  M_r / M_l, & \text{if } M_l = 0 \text{ then } M_l = 1 \\
  M_l / M_r, & \text{if } M_r = 0 \text{ then } M_r = 1 
\end{cases}
\] (1)

where \( M_l \) is the total count of the matches at the left part of the image and \( M_r \) is the total count of the matches at the right part of the image. To obtain a more robust and sensitive control equation distribution (d) might be redefined (including all matching clusters) which is able to approximate turning rate with higher accuracy, shown in the following expressions:

\[
d = \begin{cases} 
  (c_{w1} \frac{c_r}{c_l} + c_{w2} \frac{c_{mr}}{c_{ml}}), M_l < M_r \\
  c_{w1} \frac{c_r}{c_r} + c_{w2} \frac{c_{mr}}{c_{mr}}, M_r < M_l 
\end{cases}
\] (2)

where \( 0 < c_{w1} \leq 1, 0 < c_{w2} \leq 1 \)

In order to estimate the next possible turning rate (w), \( d \) is multiplied by a model parameter value, varying between 0 and 1, and can be defined as follows:

\[
w = d \times c_w, \text{where } 0 < c_w \leq 1
\] (3)

When the robot approaches the goal with the capability of keeping it in the field of view, matching strength usually tends to increase. Therefore, the matching strength can be adapted to arrange the linear velocity. In order to achieve this, a simple but efficient velocity model is proposed, illustrated in the following expressions:

\[
v = \begin{cases} 
  m_t \times k_v, & v_{th} < m_t \\
  v_t, & v_{th} \geq m_t 
\end{cases}
\] (4)

In order to constraint the linear velocity, output of the given equation \( v \) is compared with \( v_{max} \), shown in following expression:

\[
v = \min (v, v_{max})
\] (5)

where \( v_t \) = minimum linear velocity. 
\( k_v \) = constant used to convert matching value to linear velocity. 
\( m_t \) = number of total matched points. 
\( v_{th} \) = linear velocity threshold parameter. 
\( v_{max} \) = maximum accepted linear velocity.

The proposed control algorithm is designed for the vehicles in which the only interaction with the motors is carried out by using the robot’s forward speed (metres/sec) and its angular velocity (turning rate) (degree/s). At the end of each processing cycle, these two control variables completely define the output behaviours. The main idea behind the visual homing strategy is to infer the direction and/or distance to the goal location from the disparity between the current and the goal images. The control variables of the proposed algorithm can be easily adapted to a visual homing strategy with minor modifications. Homing vectors are estimated for each processing cycle until the discrepancy between the current and reference images falls below a certain threshold value. Each homing vector comprises a rotation angle to decrease the orientation difference between two images. As no metric landmark information is used, the homing vector \( \vec{H} \) is often inaccurate. The agent therefore moves by some distance (either fixed or calculated based on current sensor information) in the direction of homing vector, \( \vec{H} \). In order to make an approximation to homing vectors, Equation 2 can be used to obtain the rotation angle instead of angular velocity by changing the interval of the model parameters, shown in the following expressions:

\[
a = d \times c_{wa}, \text{where } 0.5 < c_{wa} \leq 1
\] (6)

In addition, instead of estimating linear velocity, Equation 4 can also be adapted, with a minor modification, to use to estimate forward translation (forward displacement) or the magnitude of each homing vector, illustrated in the following expressions:

\[
f_t = \begin{cases} 
  m_t \times k_{ft}, & f_{th} < m_t \\
  f_t, & f_{th} \geq m_t 
\end{cases}
\] (7)

In order to limit the forward translation, given equation \( f_t \), is compared with \( f_{t_{max}} \), shown in following expression:

\[
f_t = \min (f_t, f_{t_{max}})
\] (8)
where \( ft \) = minimum forward displacement.
\( k_f \) = constant used to convert matching value to forward translation.
\( m_f \) = number of total matched points.
\( f_{th} \) = linear velocity threshold parameter.
\( f_{max} \) = maximum accepted forward displacement.

The major difference between the conventional visual homing techniques and the proposed technique is that the proposed one is designed for monocular vision-based navigation systems. Despite the fact that the wide angle of view permits a mobile robot to interact with a curved path even around sharp corners, hairpin turns or other complicated curves, the main disadvantages of the omnidirectional vision systems are geometric distortion and poor resolution. Monocular vision is able to cope with these problems, whereas the main drawback of a monocular image acquisition system is that the target objects may be out of the field of the camera, due to its limited field of view. In order to overcome this problem and to provide reliable navigation, the SIFT-based algorithm is performed with a monocular camera equipped with pan and zoom functions, which will be discussed in the following sections of this paper.

3.2 Design of behaviours

There are several approaches to designing a behavioural-based architecture depending on the required task. In this study, the architecture has been designed based on the subsumption architecture in which each layer implements a particular goal of the robot and higher layers are increasingly abstract. Each layer’s goal subsumes that of the underlying layers [2,3]. The architecture, designed for this study, comprises five behaviours, namely: approach, collision avoiding, completed, wander and goal seeking. As opposed to many traditional AI-based approaches, the subsumption architecture uses a bottom-up design. Each behaviour will be discussed in bottom-up order respectively. A finite state machine (FSM), presenting the connections of each vision-based behaviour, is shown in Figure 3.

3.2.1 Goal seeking behaviour (GS)

Goal seeking is one of the key behaviours of these architectures and is used to find and acquire the target object in complex unknown environments. The goal seeking behaviour of the proposed architecture is designed for monocular vision systems equipped with pan and zoom functions. The main drawback of monocular vision systems, as mentioned previously, is their limited field of view, which is not appropriate for goal-based navigation in dynamic and cluttered environments. To enhance the field of view of the monocular vision system, the pan function of the camera is adapted to the navigation algorithm. The main objective of this behaviour is to seek the goal in order to compute the turning rate or rotation angle and linear velocity or forward translation of the robot on a real-time basis. The strength of the matching results, obtained from output of the algorithm, is utilized to determine the next possible state or behaviour. If the obtained matching value is higher than a predefined threshold value, ThresholdForA (Tha) or ThresholdForC (Thc), the robot enters the approach or completed state. Otherwise, the planning stage of the current behaviour is activated, which entails panning the camera left and right at a predefined (small) angle in both cases, to enhance the field of view of the robot’s vision. The most reliable way to obtain high accuracy at this stage is to pan until reaching the limitation of the physical sensor. However, the more angles the camera turns the more processing time it consumes. For real-time applications, processing time and physical limits of the camera are important limitations. Therefore, a high quality camera is essential to improve the performance in this study. For the real experiments, an interval of 490° is employed in order to obtain a high searching performance with affordable processing time based on the capacity of the vision sensor. To increase the accuracy of the navigation, a zooming stage which involves changing the focal length of the lens to bring the subject closer or further away in the frame is activated, depending on the strength of matching value. Consequently, if the consistency between the panning and zooming stages is obtained related to the goal, and the matching value is more than a predefined threshold value, approach behaviour is activated. Otherwise, if the target cannot be acquired, the wander behaviour is activated to displace the robot position randomly.

![Figure 3. FSM for vision-based behaviours](image-url)
3.2.1 Approach behaviour (A)

The fundamental aim of this behaviour is to direct the robot to its goal. This behaviour is only activated when the existence of the target is detected to a high accuracy. In order to navigate in a smooth way, the turning rate (w) or rotation angle (a) and linear velocity (v) or forward translation (t) of the robot are adjusted respectively. According to the state diagram, illustrated in Fig. 3, if the strength of the matching results is more than an appropriate threshold value, $Thc$, completed behaviour is activated. On the other hand, if the difference between the previous and current matching results is less than a predefined threshold value, $Thg$, the first panning stage is activated to seek the goal. If this fails, then progressive zooming stages are activated. If these stages succeed in tracking the goal, the robot keeps navigating in this manner until the completed behaviour is activated, otherwise the wander behaviour is activated to relocate the robot randomly, which will be discussed in the following section.

3.2.2 Wander behaviour (W)

Wandering is a form of random steering. It is the main state of the robot while it navigates within the environment and is a necessary component for navigation for any virtual, persistently populated environment. Wandering behaviour in this study is comprised of three states, namely, if the strength of the current matching is more than the previous one, but it is not sufficient to revert back to the goal seeking or approach behaviours, the control variables of the robot are recalculated and the robot keeps moving based on these values. For the second case, whenever the matching value becomes strong enough, the goal seeking or approach behaviour is activated. In the third case, if the current matching result is either zero or less than the previous one, a random search is initiated, which retains the steering direction state and makes small random displacements. For instance, at time $t_i$, the robot may be in the process of turning to the right and at time $t_i$ it will still be turning in almost the same direction. The steering vector takes a random walk from one direction to another. To produce the steering vector for the next instance, a random displacement is added to the previous value and the sum is constrained again to the sphere’s surface. The sphere’s radius determines the maximum wandering strength and the magnitude of the random displacement determines the wander rate.

3.2.3 Completed behaviour (C)

The objective is to complete the proposed task, which is activated by either goal seeking or approach behaviours, based on the strength of the matching value. When the behaviour is activated, the robot continues navigating until its goal is found, in a smooth and timely manner, by adapting its algorithm. One of the key issues with this behaviour arises as the robot approaches its destination. As the distance between the target and the actual robot position closes, the obstacle avoidance behaviour may be invoked and steer the robot to another location. As has been previously mentioned, the architecture of the system is designed based on subsumption architecture in which each layer’s goal subsumes that of the underlying layers, hence, in the case of completed behaviour activation, obstacle avoidance behaviour must be suspended, in order for the robot to reach its goal. To handle this situation, a fusion of two sensors’ data (vision and range finder) is performed by obtaining the strength of the matching value with respect to a predefined threshold value, $Threshold_{forF}(Thf)$, and estimating the distance ($d_{sg}$) between the goal and the robot using the range finder. Fusion of the vision and range finder sensors are defined as:

$$F_S = \begin{cases} 1, & m > Thf \text{ and } d_{rg} < d_{sg} \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

where $F_S$ = final stage, a boolean value.

$m$ = matching strength.

$Thf$ = threshold value for matching strength.

$d_{rg}$ = distance between the robot and the goal.

$d_{sg}$ = distance of influence of the goal.

3.2.4 Obstacle avoidance behaviour (O)

This behaviour utilizes information from the laser range finder. The perceptual schema for the output of this behaviour generates an avoiding manoeuvre for the robot relying on a neural network design. After each scan, the laser range finder returns a corresponding point for each unit of angular resolution which represents the distance between the robot and any obstacle that the laser encounters. This behaviour is activated whenever the laser range finder returns a value within a distance of influence. In this application, a 2-D laser range finder which is able to scan 240° is employed to acquire data for the network’s inputs. In principle, any kind of range finder sensor can be employed to provide input to the network, with minor modification. In order to design a neural network-based obstacle avoidance strategy, a series of left/right turns away from several obstacles are recorded, using the laser range finder. The network is trained with collected data from 320 input-output pairs. Characteristics of the network are shown in Table 1. The neural network employed in this study is a feed forward neural network with a BP learning algorithm. The input layer $n_i$ consists of eight input nodes and the output layer $n_o$ has one output. Each input of the network is associated with a 30° field of view and the mean distance value of
the field is utilized as input to the network. The avoiding vector, comprising direction and heading angle, is assigned to the output of the network. A Sigmoid function (between 0 and 1) is employed as the activation function for both output and hidden layers. The experiments in which the neural network was trained with error $10^{-5}$ have produced good results. However, as the error decreases to $10^{-6}$ and $10^{-7}$ any further improvement in the accuracy cannot be obtained and the processing time increases dramatically. According to the test results, it was concluded that the second topology, illustrated in Table 1, is the most appropriate design for avoidance behaviour. The gathered data from the sensors are normalized, based on the activation function to provide appropriate data to the network. Compared to many traditional obstacle avoidance methods, the proposed system provides a smooth avoiding manoeuvre and overcomes the local minima problem.

| Num | Topology  | Epochs | Error  |
|-----|-----------|--------|--------|
| 1   | [8-8-1]   | 227    | $10^6$ |
| 2   | [8-24-1]  | 324    | $10^5$ |
| 3   | [8-6-6-1] | 303    | $10^6$ |
| 4   | [8-8-8-1] | 426    | $10^4$ |
| 5   | [8-8-15-1]| 410    | $10^2$ |
| 6   | [8-24-5-1]| 660    | $10^5$ |
| 7   | [8-10-15-1]| 873  | $10^6$ |
| 8   | [8-10-20-1]| 1450 | $10^5$ |

Table 1. Characteristics of the training network

The obstacle avoidance behaviour is invoked and provides obstacle avoidance whenever the robot encounters any obstacles, illustrated in Figure 5. The only exception was discussed in the completed behaviour section. One of the key points, related to avoidance behaviour, is to localize the position of the robot based on the previous bearing angle after an avoiding manoeuvre, decreasing the consumed time to reach the goal. The localization technique basically attempts to locate the robot with respect to odometry readings. To incorporate odometry data into the system, a function is defined as follows:

$$\Delta_{\text{odometry}}(\Delta y, \Delta \theta)$$

(10)

where $\Delta y$ = Position difference along y-axis.

$\Delta \theta$ = Bearing angle difference.

After avoiding movement, the robot steers in the proper direction to decrease the error provided by the $\Delta_{\text{odometry}}$ function. However, the odometry readings may become increasingly unreliable over time as these errors accumulate and compound. To improve these readings, a simple bearing only measurement technique relying on monocular vision is employed with odometry readings in which the reference image and the current image are compared to enhance the localization accuracy of the robot. The vision-based localization algorithm used in this study relies on the SIFT method. According to the proposed algorithm, robust localization is performed by fusing odometry data and SIFT matching results then the matching strength between the reference image and the currently captured image is compared to the threshold value of the previously performed behaviour during the localization manoeuvre. For instance, if the matching strength obtained by the images is equal or over this certain threshold value during the localization manoeuvre, the current state is immediately interrupted and the new heading angle is generated by the control equations given in (3). On the contrary, in the case of not detecting the goal during the localization manoeuvre, which may be caused by a large obstacle in the environment, the robot keeps heading in the same direction for a while instead of activating search behaviour, which increases the sustainability of the system’s performance. An example of obstacle avoidance and localization stated are demonstrated in Figure 4.
4. Experimental Results

In order to verify the performance of the navigation system, several experiments, involving different scenarios, were implemented. The system was also evaluated using the Microsoft Robotics Studio’s (MRS) simulation environment. The results of these experiments reveal that the proposed architecture provides a safe and efficient method for indoor navigation based on a mapless navigation strategy. An example of SIFT-based matching, used for navigation task, is illustrated in Figure 6.

4.1 Simulation results

The system was implemented in a MRS simulation environment for vision-based navigation. Due to its flexibility, different scenarios involving different environments and one or more robot models can be employed. In these experiments, a Corobot robot model, mobile robot platform designed to minimize the complexity of robot development, is utilized. In order to evaluate the performance of the whole system, several tests with respect to different scenarios were implemented. One of these scenarios is discussed in this section as follows:

Scenario 1: the mobile robot is required to navigate two different goals respectively in a partially cluttered environment, shown in Figure 7, and avoids unexpected obstacles located along its path as shown in Figure 8. For the first goal it navigates from the ‘Start Position’ (4, 0, 5) to the ‘Goal Position’ (3.3, 0, -4.0) and from there it navigates to the ‘Second Goal Position’ (-3.3, 0, 0.59). Matching strength with respect to the travelled distance is illustrated in Figure 9.

Figure 6. An example of SIFT matching between the reference and goal images during navigation task.

Figure 7. Scenario 2 for simulation environment

Figure 8. Estimated trajectory for scenario 2

While the robot approaches the target it loses contact with the goal with respect to the obstacle (placed in front of the goal) and therefore search behaviour is activated to seek the goal. However, it fails to detect the goal for the same reason.
Figure 9. Evaluation of Scenario 1, (a) Matching strength for the first goal during the navigation task, (b) Matching strength for the second goal during the navigation task.

Ultimately, Wander behaviour is activated to enlarge the robot’s field of view which finally accomplishes perceiving the goal, which unexpectedly in this case consumes additional processing time. The robot then resumes its path and continues moving to achieve its goal while avoiding to the second obstacle with a smooth manoeuvre. The results of these simulation based experiments present that the proposed architecture is able to safely navigate the robot to a single or associated goals in particularly cluttered environments. The robot was able to navigate towards the goal while avoiding collision with any walls or obstacles and did not damage itself or to the other objects when it navigated throughout the test environment.

4.1 Real experiments

The system was developed on a Pioneer 3-DX mobile robot, with an on-board Intel Pentium 1.8 GHz (Mobile) processor, and includes 256 Mbytes of RAM memory, as shown in Figure 10. The mobile robot used in this study has been developed as a part of the Intelligent Robot Swarm for Attendance, Recognition, Cleaning and Delivery (IWARD) project, and is equipped with a Hokuyo URG-04LX laser rangefinder. An Axis-213 Internet-based pan-tilt-zoom camera, 25 frame rate, was integrated into this system. The software architecture of the proposed system is supported by Player Architecture, which are open-source software projects. All experiments were conducted in an area of the Robotics and Automation Research Laboratory of Newcastle University, which has physical dimensions of 15.60m x 17.55m, as illustrated in Figure 11, and the corridor, in front of the this lab, having physical dimensions of 17m x 12m. To evaluate the performance of the system, several different scenarios were performed and two of them will be discussed in this section. System parameters for the experiments are illustrated in Table 2. 352x288 resolution is just utilized for goal seeking behaviour, however for navigation 176x144 resolution is used. Comparison of the enhanced SIFT, powered with OpenMP[15], and the conventional SIFT algorithms is illustrated in Figure 16. The algorithms are applied to four different goals, as illustrated in Fig 17. The OpenMP library provides a basic script language which basically facilitates to parallelize code [15].

Figure 10. Pioneer 3-DX robot with Axis 213

Figure 11. Test Environment

Figure 12. Estimated trajectory for Scenario 1
Scenario 1: during the first set of experiments, a static obstacle is located in the experimental environment and the robot is required to travel from its start position to a goal position while trying to avoid obstacles. An example illustrating this scenario is shown in Figure 12. The goal is detected using the goal seeking behaviour and the robot is guided toward the goal. The robot maintains navigation until encountering an obstacle within its environment. In order to avoid the obstacle, the robot makes a clear turn to the left and performs an in-place rotation toward the goal to nullify the heading error based on the employed localization techniques. Once the robot is aligned with the goal direction, it then proceeds towards its goal whilst avoiding the second obstacle with right turn manoeuvres. Generated control variables during the task are illustrated in Fig. 13. This experiment reveals that the robot is able to perform the task with a high degree of success, in the case of static obstacles.

Scenario 2: The second experiment is conducted in the corridor environment, illustrated in Figure 14, in which the robot is required to navigate towards two different goals respectively. In this experiment, goals are associated with each other that the robot knows to turn 90° to right after completing the first goal. The first goal is detected with zooming state at the start position, the robot, therefore, navigates towards the goal until it perceives the obstacle. The robot avoids the first obstacle successfully and localizes itself with respect to the previous position (the position just before the obstacle avoidance behaviour is enabled). After which it continues moving towards the goal whilst avoiding the second obstacle with a smooth manner. As it reach the first goal, it turns to left and once the second goal is detected the robot moves towards the goal. This experiment shows that the robot is capable of steering multiple goals whilst avoiding unexpected obstacles in any corridor environment. Generated control variables for this scenario are shown in Fig 15.

![Figure 13. Generated control variables during the task](image1)

![Figure 14. Generated control variables during the task](image2)

![Figure 15. Generated control variables during the task, (a) For goal 1, (b) For goal 2](image3)

| Parameter                  | Value  |
|----------------------------|--------|
| $d_{ig}$ (distance of influence of the goal): | 0.5 m  |
| $v$ (minimum speed):        | 0.1 m/s|
| $v_{max}$ (maximum speed):  | 0.5 m/s|
| $c_s$ (smoothing parameter):| 0.8    |

Table 2. Control parameters
5. Conclusions

This paper describes a novel behaviour-based architecture for mobile robot image-based navigation. The architecture draws its inspiration from many different disciplines and comprises several layers with each level responsible for different tasks. The highest level is based on a visual homing strategy, employing a SIFT algorithm, which is accepted as the most reliable feature extraction technique to detect landmarks. Performance of the classical SIFT algorithm is not considered suitable for real-time applications and has been enhanced by employing OpenMP, supporting multi-platform shared memory multiprocessing. An original control algorithm, involving a SIFT-based visual homing strategy, relying on a single camera has been proposed. The control algorithm basically employs the strength and distribution of the matched points, which does not require any camera calibration and complex mathematical calculations. One of the important layers is the obstacle avoidance layer, which is designed around a laser range finder. The obstacle avoidance module of this architecture in this study is realized using a multilayer neural network, which provides smooth and safe avoiding manoeuvres. In order to verify the system, all experiments were conducted in appropriate indoor environments and a complex scenario was performed in a powerful 3-D simulation environment involving different environments and conditions. As a result, these experiments have validated that the proposed system is capable of controlling an autonomous robot in unknown and uncertain environments, directing it to any kinds of goal while avoiding obstacles. Future work will be devoted to adequately integrating the global positioning system (GPS) to the proposed architecture to provide a fully autonomous navigation system for an outdoor environment.

Figure 17. An image database, including arbitrary images, is used to evaluate the performance of enhanced SIFT algorithm.

6. References

[1] Althaus, P. (2003) 'Indoor Navigation for Mobile Robots: Control and Representations', PhD thesis, Stockholm University.
[2] Brooks, R. A. (1986) 'Robust Layered Control System for a Mobile Robot', IEEE Journal of Robotics and Automation, RA-2, (1), pp. 14-23.
[3] Hao, L. and Yang, S. X. (2003) 'A behavior-based mobile robot with a visual landmark-recognition system', Mechatronics, IEEE/ASME Transactions on, 8, (3), pp. 390-400.
[4] DeSouza, G. N. and Kak, A. C. (2002) 'Vision for mobile robot navigation: A survey', IEEE Transactions on Pattern Analysis and Machine Intelligence, 24, (2), pp. 237-267.
[5] Ohyu, I., Kosaka, A. and Kak, A. (1998) 'Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing', Robotics and Automation, IEEE Transactions on, 14, (6), pp. 969-978.
[6] Chen, Z. & Stanley, T. (2006), 'Qualitative Vision-Based Mobile Robot Navigation', IEEE International Conference on Robotics and Automation (ICRA), 2686-2692.
[7] Szenher, D. M. (2008), 'Visual Homing in Dynamic Indoor Environments', PhD thesis, University of Edinburgh.
[8] Remazeilles, A. and Chaumette, F. c. (2007) 'Image-based robot navigation from an image memory', Robot. Auton. Syst., 55, (4), pp. 345-356.
[9] F. Vassallo, R., Schneebeli, H. J. and Santos-Victor, J. (2000) 'Visual servoing and appearance for navigation', Robotics and Autonomous Systems, 31, (1), pp. 87-97.
[10] Lowe, D. (2004) Distinctive image features from scale-invariant key points, International Journal of Computer Vision, 60-2, 91-110
[11] Mikolajczyk, K. and Schmid, C. (2005) 'A performance evaluation of local descriptors', Pattern Analysis and Machine Intelligence, IEEE Transactions on, 27, (10), pp. 1615-1630.
[12] Vardy, A. & Oppacher, F. (2003) ‘Low-level visual homing’, In Advances in Artificial Life: Proc. of the Europe Conf. on Art. Life.

[13] Se, S. Lowe, D. And Little, J. (2001) ‘Vision-based mobile robot localization and mapping using scale-invariant features’, Proceedings 2001 ICRA. IEEE International Conference, (2), pp. 2051-2058.

[14] Theil, E. (2000) ‘Networks and artificial intelligence for biomedical engineering [Book Review]’, Engineering in Medicine and Biology Magazine, IEEE, 19, (5), pp. 73-91.

[15] Ayguade, E. Copty, N. Duran, A. Hoeffer, J. Yuan, L. Massaioili, F. Teruel, X. Unnikrishnan, P. and Guansong, Z. (2009) ‘The Design of OpenMP Tasks’, Parallel and Distributed Systems, IEEE Transactions on, 20, (3), pp. 404-418.