Hierarchical Fuzzy State Controller for Robot Vision

Donald Bailey[1], Yu Wua Wong[1], and Liqiong Tang[2]

[1] Institute of Information Sciences and Technology,
[2] Institute of Technology and Engineering
Massey University, Palmerston North, New Zealand
D.G.Bailey@massey.ac.nz, yuwua.w@gmail.com, L.Tang@massey.ac.nz

Abstract. Vision algorithms for robot control are usually context dependent. A state based vision controller can provide both the computational and temporal context for the algorithm. A hierarchical layering enables one or more sub-objectives to be selected. To mimic human behaviour, it is argued that using fuzzy logic is better able to manage the subjective data obtained from images. Fuzzy reasoning is used to control both the transitions between states, and also to directly control the behaviour of the robot. These principles are illustrated with an autonomous guide robot. Preliminary results indicate that this approach enables complex control and vision systems to be readily constructed in a manner that is both modular and extensible.

Keywords: vision system, service robot, image processing, fuzzy control.

1 Introduction

Vision is an important component of modern robotic systems. This is even more the case with mobile and autonomous robots because vision is the most sophisticated way of perceiving the environment [1,2]. As each type of robot has different functionality and different objectives, there is a wide range of different vision systems and there is no common method or technique to build vision systems for such robots. As computer vision methods are highly adapted to meet each robot's objectives, there are various adaptations to each vision system. The image processing methods used for each vision system can differ, but the aims of the vision systems usually coincide. There are at least four broad areas or purposes for which robot vision is used:

1. *Navigation and path planning*: Vision can be used to determine the location of obstacles and clear regions with the purpose of navigating a suitable path through the environment. This is particularly important if the environment is unknown (to the robot), or dynamic with other objects moving.
2. *Location and mapping*: Closely related to navigation is building and maintaining a map of the environment. Here, vision can be used to identify landmark features (both known and previously unknown) and by maintaining a database of such features (the map) the robot location may be determined through triangulation and ranging [3]. An existing map can be extended by adding the location of new landmarks relative to those already in the database.

G. Sommer and R. Klette (Eds.): RobVis 2008, LNCS 4931, pp. 391–402, 2008.

3. *Visual servoing*: Vision is a valuable sense for providing feedback to a robot when performing its designated task [4]. When applied to a mobile robot itself, this comes under the previous two purposes. However, visual servoing also enables manipulators to be positioned accurately relative to the object being manipulated, and can be used to facilitate alignment. Thus vision can be used to enable a robot to achieve an improved precision over that which might otherwise be achieved.
4. *Data gathering*: The role of many autonomous mobile robots is to gather information about the environment in which they are placed. Such data gathering may be the primary purpose of the robot, or may be secondary, enabling the robot to achieve its primary task.

For an autonomous robot to be practical in most situations, its movement must be continuous and at a sufficient speed to complete its given task in an acceptable time period [2]. Hence, there is often an emphasis on maximising image processing speed, while simultaneously reducing the size, weight, and power requirements of the vision system. Such constraints require simplifying the image processing where possible, without compromising on the functionality of the system.

Interest in intelligent robot systems has increased as robots move away from industrial applications and towards service areas such as personal service in home and office, medicine, and public services [5]. In the service sector, there are many fields in which autonomous robots can be applied in; it can range from medical to fast food to home and domestic, as there are a variety of services and tasks that is unappealing or dangerous for humans, but are suitable for robots. For a truly autonomous robot to fulfil the necessary tasks in any service environment, it must be able to interact with people and provide an appropriate response. Hence, many robots have been designed to mimic human behaviour and form.

2 Autonomous Guide Robot

Currently, we have an ongoing project to build an autonomous robot guide system, which aims to provide services for searching in an unknown office environment [6]. Given a name, the robot needs to be able to enter a corridor, search for office doors, and locate the office associated with the name. In terms of the broad task defined above, such a robot needs to be able to perform a number of sub-tasks:

1. *Navigation and path planning*: The robot needs to be able to move down the centre of the corridor, while avoiding people and other obstacles within the corridor.
2. *Location and mapping*: As the robot is searching, it builds an approximate "map" of the office environment. This will facilitate later searches along the same corridor.
3. *Visual servoing*: While this is related to navigation, the robot must be able to position itself appropriately relative to each door to enable the nameplate to be effectively and efficiently read.
4. *Data gathering*: As the main task is the location of a particular office, the data gathered at each step is the name on each office door.

The image processing requirements of each of these tasks are context dependent. How an image is processed depends on the particular task being performed, which in turn depends on which step the robot is at within its overall goal. While the image processing requirements for each individual sub-task are straight forward, we would like the system to be able to be extended easily as new tasks are added to the robot system. Without this as an explicit design requirement, there is a danger that the resultant vision system would be monolithic, making it difficult to adapt to new tasks.

These objectives may be achieved by using a state-based controller [7]. A finite state machine is used to encapsulate the current state of the robot and vision system, effectively providing the context for the current sub-task. The current state represents the particular step being performed by the robot at the current time. As such, it is able to specify the particular vision algorithms that are relevant to the current step. This is the computational context of the particular step. The state machine can also be augmented with any historical data that has been gathered to provide the temporal context. This would enable it to effectively integrate the secondary mapping tasks with its primary goal of finding a particular office.

Such a state based approach enforces a task oriented modular design on both the vision system and robot control. It also decouples the particular vision algorithms associated with each step from the control tasks, which are encapsulated within the state machine. Adding a new task consists of updating the finite state machine, or adding a new state machine that breaks the new task down to atomic steps. For each step, the vision algorithms are very specific, making them easier to design or develop.

As many tasks can be broadly specified, and broken down into a number of simpler steps, or sub-tasks, it is convenient to use a layered or hierarchical approach to the design of the individual state machines [7,8,9,10]. By breaking down a complex task into a series of simpler tasks, the components or sub-tasks that recur may be encapsulated within and represented by separate state machines. This enables re-use of common components within a different, but related task. Such a breakdown also enables multiple concurrent tasks to be performed simultaneously, for example maintaining the position in the centre of the corridor, obstacle avoidance, searching for doors, and mapping. With parallel sub-tasks, it is important to be able to prioritise the individual parallel tasks in order to resolve conflicting control outputs.

2.1 Mimicry of Human Behaviour

Another principle we used in the design of the guide robot system was to mimic human behaviour [6], particularly with regards to the way in which a person searches for an unknown office.

If a person has already visited the office previously, they remember the approximate location of the office. This enables them to proceed directly in the right direction, using vision to confirm previously remembered landmarks and verify that they have located the correct office. In a similar manner, our robot system will use a previously constructed map if it recognises the name. Such a map can speed the search because not every door would need to be checked. The first door or two can be used to validate the previously constructed map, and the task then becomes a simpler one of map following rather than searching. When the destination is reached, the nameplate on the door is located and read to verify that the correct office has been found.

On the other hand, when a person enters an unknown corridor or building, they randomly choose a direction to move in. Then, they will move in the chosen direction and search for doors. At each door, they will read the nameplate and decide whether it is the correct office depending on which office they are searching for. Similarly, the robot system will move through a corridor, search for doors, read the nameplate of each door, and decide whether it is the correct office. At the same time, the robot can be constructing a map of office locations along the corridor to speed later searches.

One of the characteristics of the human vision system is that it does not consciously look too deeply into the details. This is in contrast to conventional machine and robot vision systems. For example a human identifies a door as a whole because of its general characteristics (e.g. it is rectangular, it has a nameplate and door handle, etc), and gauges it to be similar to a door. The human vision system also does not automatically calculate distance of the door from its relative position. Instead, it will gauge whether the door is far away or close, and whether it is on the left side or right side.

In contrast, conventional vision systems and associated robot control tend to deal with details, and in particular with numerical details of data extracted from the image. It could be argued that much of this numerical precision is artificial and is not necessary for many control tasks. Everything, including any robot control variable, is calculated with precision according to the particular models used. In many cases however, and this applies more so with computer vision, the models usually represent only a simplified approximation of reality and are therefore not as precise as the calculations imply.

Fuzzy logic provides a way of managing such imprecision, and in particular provides a mechanism for reasoning with imprecise data in a way that more closely resembles the way in which humans reason about imprecise and subjectively defined variables.

2.2 Fuzzy Logic

Fuzzy sets are generalised from conventional set theory, which was introduced by Zadeh in 1965 [11]. It is a mathematical way to represent the subjectiveness and imprecision of everyday life. Even when quantities can be measured precisely, the subjective interpretation of that value is often imprecise, and may well be context dependent. A quantitative measurement may be mapped to qualitative or subjective description (a fuzzy set) through a membership function. Such membership functions are often overlapping and provide an interpretation of the quantity.

Fuzzy logic is derived from fuzzy sets, and provides a mathematical basis for reasoning or deriving inferences from fuzzy sets based on rules. Each rule consists of a set of conditions and produces an inference. Fuzzy operators are used to evaluate the simultaneous membership of the conditions, with the resulting membership evaluation associated with the inference. All applicable rules are evaluated, with the results combined to select one or more actions. A crispening function is usually applied to convert the fuzzy output to a binary membership or numerical control variable for the action.

Within robotics, fuzzy logic can be used for any of the control activities (see for example [12,13]). We are using it to both select the next sate within the associated state machine, and also to select the robot actions to perform within the current state. This flow is illustrated in fig. 1. The current state within the state machine associated with a particular sub-task provides the context for this activity. Therefore the current state specifies (in practice, selects from a library) both the image processing algorithm to apply to the input, and the fuzzy rule set to be used. Any numerical outputs from the vision processing are fuzzified, as are any other relevant inputs to the current processing such as camera angle, etc. The fuzzy rules are applied to the current inputs with the results used to drive any actions to perform by the robot in the current state, and where necessary to select the next state.

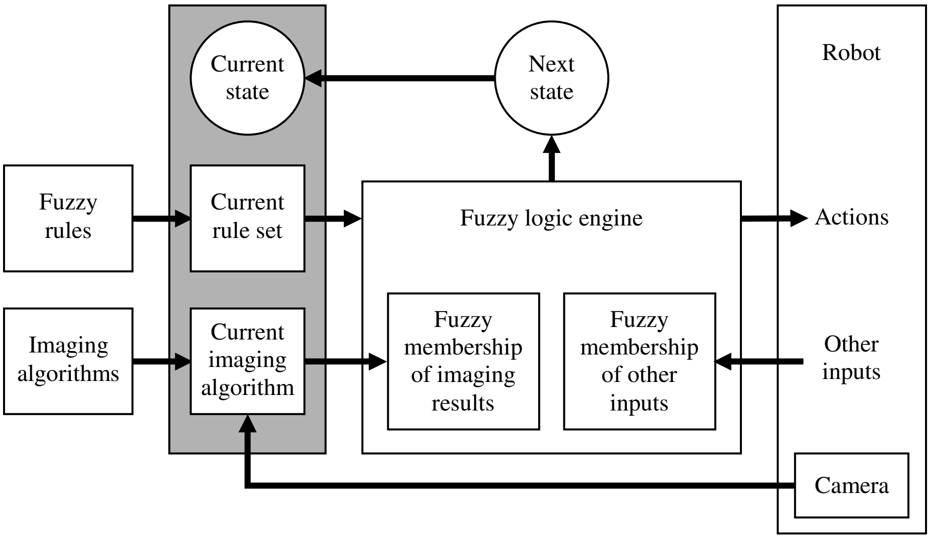

Fig. 1. The working of the fuzzy state control: the current state selects the rule set and imaging algorithm; these are applied to the camera image and other inputs to generate the next state and the robot action

3 System Design

3.1 Imaging System

The system is designed to use lightweight and relatively low cost firewire camera to facilitate interfacing to the control computer (a lightweight laptop) on the robot. Such cameras tend to be low resolution, typically 640 x 480 pixels, and usually have a moderate angle of view. When the robot is positioned approximately in the centre of the corridor, it needs to be able to easily and unambiguously read the nameplate on the door. To clearly resolve the individual letters requires a resolution of approximately 1 mm per pixel on the nameplate. While the resolution can be slightly less, this provides a margin to allow for inevitable positioning errors of the robot within the corridor. However, at this resolution, a typical office door (which is about 0.8 m wide) does not fill the complete field of view.

Our first thought of detecting the four edges of the door (left, right, top and bottom) to use these to detect perspective to give the relative position between the robot and the door is therefore impractical. Unless the door is viewed obliquely, there is insufficient resolution to view both sides of the door simultaneously, let alone detecting the angle of the door top and bottom. Also, at such an angle, even if the perspective could be corrected, the resolution of the lettering on the nameplate makes it impractical to read reliably.

In order to turn the limited field of view to our advantage, we need to realize that we are not just processing a single image of each door, but are able to capture many images as to robot moves past. By recognizing key features within the image, we are able to use the robot motion, combined with panning of the camera to servo the robot into the correct position for reading the nameplate. The state machine is then able to record the context consisting of the previous steps leading up to the current step, allowing us to focus the image processing on a single step at a time. This also enables us to simplify the image processing, because we do not need to measure with high precision, and do not necessarily require the whole object within the field of view to make useful inferences.

3.2 Robot Capabilities

Because the camera is on a mobile platform, it is necessary to take into consideration the actions that the robot may perform to modify the viewpoint. This is particularly important because we are using visual servoing to position the robot appropriately for reading the doors.

The robot itself is wheeled, and is able to move forwards or backwards. Steering is accomplished by differentially driving its wheels. A camera is mounted on the robot, on a pan-tilt base. This allows the camera angle to change as the robot moves, and enables the robot to read the nameplates without having to physically stop and turn the whole robot. The scanning camera is mounted at a height of approximately 1.4 m above the ground. This is sufficiently high to enable an image of the nameplate to be captured free of perspective. By having the camera lower than that of a typical nameplate, perspective cues may also be used to assist servoing. Under consideration is the use of a second, fixed camera looking forward. This will aid in navigation, and will save having to continuously pan the search camera for this purpose.

3.3 Door Search State Machine

Fig. 2 shows our state machine for locating and reading the door nameplates. The state machine entry conditions assume that the robot is facing along a corridor, positioned approximately in the centre. A separate state machine is used to provide the initial positioning of the robot, and also to maintain the robot approximately centred in the corridor as it travels. Separate state machines are also used to negotiate corners and other obstacles.

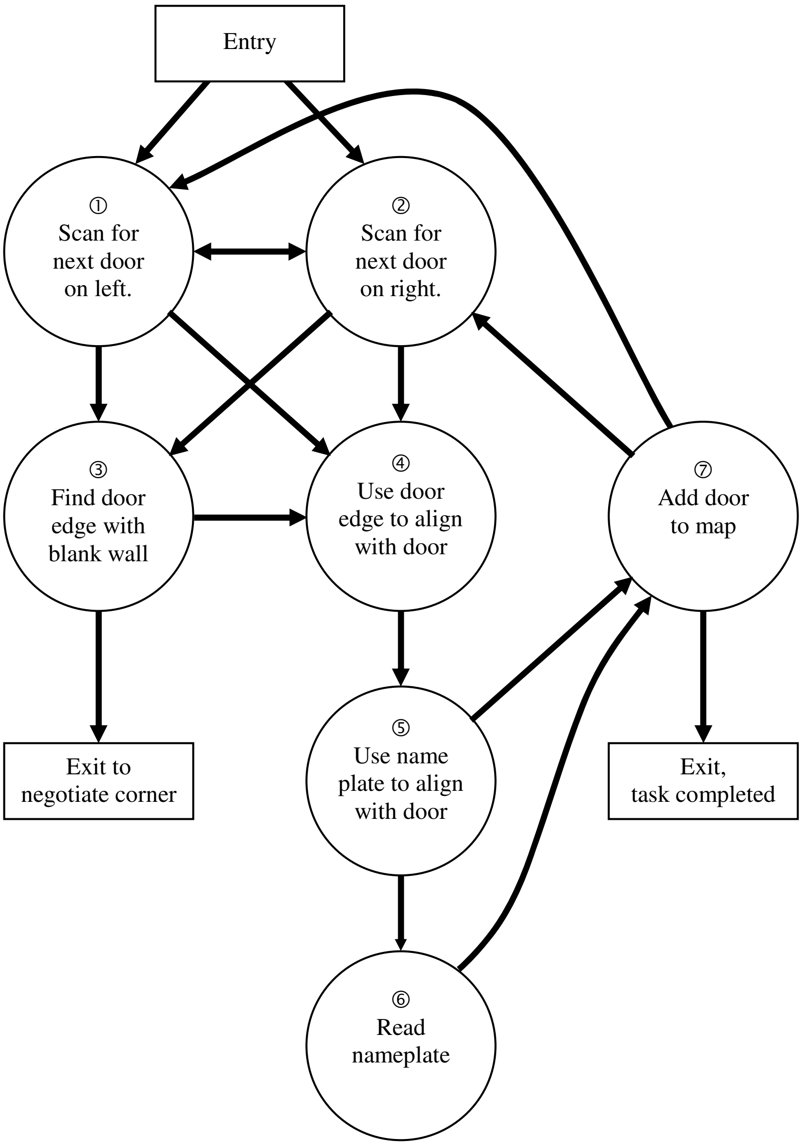

Fig. 2. State transition diagram for searching for a door. First the door edge is detected and this is used to align the robot approximately. Then the nameplate is detected and used to position the robot more carefully. Once positioned, the nameplate can be read.

The first step is to scan the corridor looking for doors along the left and right walls. Which of the two states is entered first depends on the initial orientation of the camera. The primary focus of this scanning step is not to map the corridor, but to determine on which side of the corridor the nearest door is located. Note also that the state machine is primarily action and processing focused. Additional state information is also maintained that is not directly apparent in the state machine. For example, after first scanning the left side of the corridor for the nearest door (state ①), the right side is then scanned (state ②). The approximate position of any door found in state ① is maintained in state ②.

To correctly read the nameplate, the robot must be positioned approximately perpendicular to the door. The first step in accomplishing this is to use the edge of the door to position the robot in such a way that it can obtain a clear view of the nameplate (state ④). The nameplate is then used to position the robot such that it can read the nameplate reliably (state ⑤). Precise alignment is not critical in this application; there is a relatively broad range of positions where the nameplate may be read.

Splitting the processing in this way greatly simplifies the image processing tasks. In state ④, the critical aspect is detecting the edge of the door. At a moderate to close range, this edge will fill all of the field of view. A relatively simple contrast enhancement followed by a linear vertical edge detector is sufficient to detect the door. There may also be additional edges from objects such as notice boards, or shadows from lighting in the corridor. These may be distinguished by placing limits on the linearity of the edges, the length of the edge, and the hue contrast across the edge. The context is used to distinguish between the near and far edges of the door.

The goal of this step is to position the near edge of a door on the left side of the corridor within the left edge of the image (and the converse for doors on the right side of the corridor). This will ensure that the nameplate will be completely visible within the image. In state ⑤ the focus shifts from the door edge to the nameplate. At this stage, most of the image should be filled with the door. The nameplate provides a strong contrast against this, making detection relatively easy; the nameplate is identified as a rectangular object within specified area, aspect ratio, and angle limits. These limits eliminate any miscellaneous objects such as door notice boards and other "decorations" from consideration. If the door is unlabelled, wide open, or the nameplate is unable to be distinguished for other reasons, the robot abandons this door (progressing to state ⑦). Otherwise the angle of the nameplate, and its position within the field of view are used to position the robot in the centre of the door in front of the nameplate.

The high contrast of the letters on the nameplate, combined with the elimination of skew and perspective by robot positioning, makes the nameplate relatively easy to read. A combination of syntactic pattern recognition and fuzzy template matching are used for optical character recognition. The nameplate is parsed with the individual line segments that make up each character are compared with a preset template. If necessary, multiple successive images of the nameplate may be parsed from slightly different viewpoints (as the robot moves past the door), with majority voting used to improve the reliability of the recognition.

3.4 Membership Functions and Fuzzy Rules

Most of the image processing steps do not require great precision. For example, the position of the edge of the door within the image is used primarily as a cue for detecting the nameplate. Similarly, the position of the centre of the nameplate within the image, combined with the pan angle of the camera relative to the robot, is used for alignment. Since the camera is lower than the nameplate, if the robot is not directly perpendicular to the nameplate, it will appear on an angle within the image. If necessary, the camera can be tilted further to further exaggerate this effect. The fuzzy membership functions for these variables are illustrated in Fig 3. (Note these are only some of the variables used).

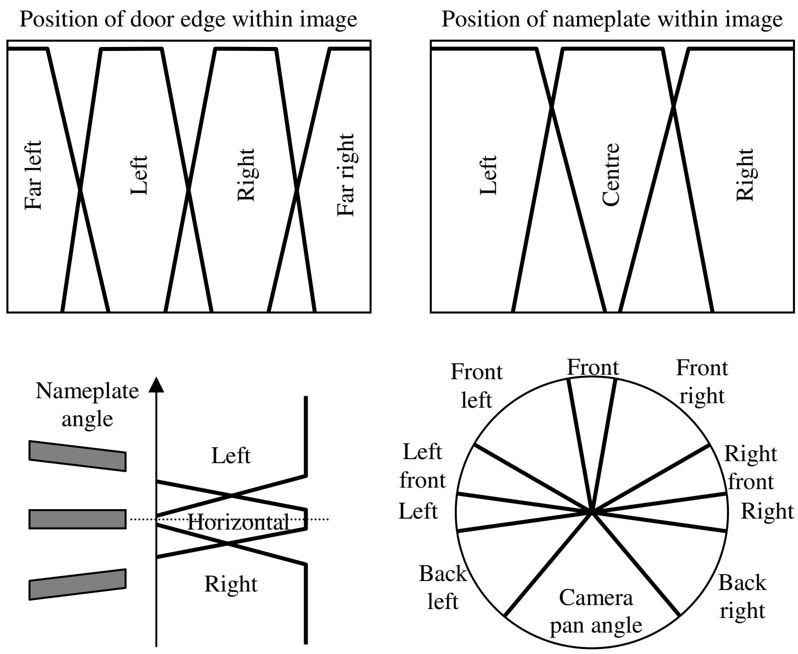

Fig. 3. Graphical illustration of some of the key fuzzy membership functions

As indicated previously, there are two types of rules: those that result in robot actions, and those that result in state transitions. Example of some of the rules used within state ④ are given here:

```
IF door_edge IS far_left
AND pan_angle IS left_front
THEN SET_STATE = 5

IF door_edge IS left
AND pan_angle IS front_left
THEN ACTION robot(forward,fast)

IF door_edge IS left
AND pan_angle IS left_front
THEN ACTION robot(forward,medium), ACTION pan(left)
```

All of the rules are evaluated, with the AND operation selecting the minimum of the corresponding membership values, and an OR operation selecting the maximum. A NOT subtracts the membership value from 1. The derived membership value from the rule is then associated with the rule's action or consequence. Where multiple rules give the same action, the results are ORed.

To derive a control action, each fuzzy output class is associated with a corresponding crisp numerical control value. Where there are multiple conflicting actions, for example move forward fast and move forward slow, the associated crisp control values associated with each class are weighted by the calculated membership values to determine the final control value.

In the case of state transitions, any transition with a membership value less than 0.5 is discarded. If multiple conflicting transitions result, then the transition with the largest membership value is selected. Ties are broken by the precedence determined by the order that the rules are evaluated, with earlier rules given a higher priority. This simplifies the programming as only one state transition needs to be retained while evaluating the rules. The "next state transition" is initialised with the current state with a fuzzy membership of 0.5. As state transition rules are evaluated, any that give a higher membership function for a transition overwrites the previous transition.

4 Results

The concept of fuzzy state control was tested by implementing using MATLAB the rules and vision algorithms for the state machine in the section. For input, a large number of images along a corridor were taken at several different camera angles. From a particular view, the fuzzy reasoning was applied to determine the actions to take. These actions were then used to select the next view within the sequence. While the spacing between the views is limited, only a small range of pan angles is available, and the temporal resolution is considerably lower than that on a real robot, this was sufficient to check that the robot could servo onto the door location and successfully read the nameplate. It also enabled the rule set to be tested and debugged for each combination of input variables – something that cannot easily be accomplished when running in real time with the robot.

Our initial tests have verified that doors within corridors can be identified, and the position of the door can be "measured" relative to the robot position, in a similar manner to human vision.

At this stage, we only have the door location state machine developed. For a fully working system, it is also necessary to develop the state machines that maintain the robot path (including obstacle avoidance), and negotiate corners. The mapping algorithms also need to be further developed. In the case of mapping, it is not necessary to create a precise map; it is sufficient to record the sequence of doors along the corridor with approximate measures of distance (such as can be derived from a sensor on the robot wheel).

The next step is to implement the system on a physical robot and test the hierarchical fuzzy state controller in practice.

5 Conclusions

A robot vision system using fuzzy logic control to identify object in real time is presented. The system is able to search and detect doors and door position as the robot navigates within a corridor. The research contributions of this paper are the concept of using a state based system to maintain context, and combining it with a fuzzy logic controller to mimic the low level, imprecise reasoning of humans in order to simplify the vision tasks for a mobile service robot. The use of fuzzy logic means that the data extracted from the images does not need to be precise, enabling less complex image processing operations and algorithms to be employed. This enables real-time operation on a relatively low-power computing platform. We have demonstrated a proof of concept of this approach for object searching and identification, through locating a specified office along a corridor.

While only a single sub-task is described in detail in this paper, we believe that this concept can be readily extended to more complex tasks through the use of multiple concurrent state machines. A hierarchical approach would allow separate state machines to be developed for common tasks, with one state machine effectively calling another to handle those tasks. This enables code reuse, and provides an extensible mechanism by which complex tasks may be decomposed.

To optimise the design of such vision systems that mimic the behaviour of the human vision system requires more in-depth research. However this paper outlines the key concepts and illustrates these with a simple example.

References

1. Cokal, E., Erden, A.: Development of an image processing system for a special purpose mobile robot navigation. In: Proceedings of the 4th Annual Conference on Mechatronics and Machine Vision in Practice, pp. 246–252 (1997)
2. Barnes, J., Liu, Z.Q.: Embodied computer vision for mobile robots. In: 1997 IEEE International Conference on Intelligent Processing Systems, pp. 1395–1399 (1997)
3. Spero, D.J., Jarvis, R.A.: On localising an unknown mobile robot. In: The Second International Conference on Autonomous Robots and Agents, pp. 70–75 (2004)
4. Jafari, S., Jarvis, R.: Relative visual servoing: a comparison with hybrid intelligent models. In: The Second International Conference on Autonomous Robots and Agents, pp. 328–332 (2004)
5. Asada, T., et al.: Developing a service robot. In: Proceedings of the IEEE International Conference on Mechatronics & Automation, pp. 1057–1062 (2005)
6. Wong, Y.W., Tang, L., Bailey, D.G.: Vision system for a guide robot. In: 4th International Conference on Computational Intelligence Robotics and Autonomous Systems (2007)
7. Sen Gupta, G., Messom, C.H., Sng, H.L.: State Transition Based Supervisory Control for a Robot Soccer System. In: 1st IEEE International Workshop on Electronic Design Test and Applications, pp. 338–342 (2002)
8. Brooks, R.A.: A robust layered control system for a mobile robot. IEEE J. of Robotics and Automation 1, 1–10 (1986)
9. Ahmad, O., et al.: Hierarchical, concurrent state machines for behavior modeling and scenario control. In: Proceedings of the fifth Annual Conference on AI, Simulation and Planning in High Autonomy Systems, pp. 36--41 (1994)
10. Sen Gupta, G., Messom, C.H., Demidenko, S.: State transition based (STB) role assignment and behaviour programming in collaborative robotics. In: The Second International Conference on Autonomous Robots and Agents, pp. 385–390 (2004)

11. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)
12. Wang, Y.L., et al.: Intelligent learning technique based on fuzzy logic for multi-robot path planning. J. of Harbin Institute of Technology 8(3), 222–227 (2001)
13. Shahri, H.H.: Towards autonomous decision making in multi-agent environments using fuzzy logic. In: Mařík, V., Müller, J.P., Pěchouček, M. (eds.) CEEMAS 2003. LNCS (LNAI), vol. 2691, pp. 247–257. Springer, Heidelberg (2003)