Autonomous Robots 10, 251–266, 2001

Supervised Autonomy: A Framework for Human-Robot Systems Development

GORDON CHENG*,† AND ALEXANDER ZELINSKY
Department of Systems Engineering, Research School of Information Sciences and Engineering, The Australian National University, Canberra, ACT 0200, Australia, http://syseng.anu.edu.au/rsl/
gordon@etl.go.jp
alex.zelinsky@anu.edu.au

Abstract. In this paper we present a paradigm for robot control, *Supervised Autonomy*. *Supervised Autonomy* is a framework, which facilitates the development of human robot systems. The components which this framework embraces has been devised in a human-oriented manner, to augment users in accomplishing their task. The general concept of our paradigm is to incorporate supervisory control with a qualitative approach for the control of robots. Supervisory control does not rely on human users to perform all the basic functions of perception and action in a system. The approach we have taken shifts all basic autonomous functions to the physical robot agent, integrated with a set of qualitative instructions, in combination with a simple graphical user interface, and together with suitable feedback form the complete framework. Experimental results of applying this framework to the use of a mobile robot teleoperation system are presented. The system we have developed make extensive use of behavior-based control technology, embracing a number of real-time visual behaviours, together with a set of intuitive instructions designed for the navigation of a mobile robot.

Keywords: supervised autonomy, mobile robot navigation, teleoperation, human robot interface, visual navigation, behavior-based control, visual behaviors, supervisory control

1. Introduction

In the opening pages of the published book "The Robotics Revolution", Scott writes, "... at the end of the day, however wonderful the robots, it is the *humans* who mean the most." (Scott, 1984). Very few people would not share this view. Historically, robotic research has been oriented towards the development of systems that can assist people to perform their tasks. Today, little has changed in orienting robotics towards this goal. This, and the large number of domains in which autonomous mobile robots can be applied, has motivated us to investigate a framework that can facilitate the interaction between people and robots. We consider that, any proposed methodology should take into consideration the alleviation of stress on human users while providing suitable level of instructions and feedback. The term "Alleviating stress" means that a human user need not be burdened with the complete control of a system. "Suitable instructions" means that a human user should not need to know the complex instructions required to command a robot to perform a task. "Suitable feedback" means that feedback is given both in appearance and explanation of what have taken place, must also be provided.

With these considerations we propose a methodology of "Supervised Autonomy", that provides a framework for the construction of human-robot interactive

*The work reported was conducted at the Australian National University.

†Present address: Humanoid Interaction Laboratory, Intelligent Systems Division, Electrotechnical Laboratory, 1-1-4 Umezono, Tsukuba, Ibaraki 305-8568. Japan.

systems. It is consists of five interconnecting components: "*self-preservation*", "*instructive feedback*", "*qualitative instructions*", "*qualitative explanations*", and "*user interfaces*".

"*Self-preservation*"*—How should we alleviate the burden placed on the user?*
The idea of "*self-preservation*" is to shift the general control of the robot from the *user* back to the robot itself. "*Self-preservation*" controls include safety aspects of the robot, such as collision and obstacle avoidance. It removes the need for a tight command loop controlled by the user. For example, if the robot has been instructed to visually servo to a selected target, and unexpected obstacle appears, the robot will move around it or stop to avoid collision, and then attempt to continue servoing.

"*Instructive feedback*"*—Do you see what I see?*
The notion of "*Instructive feedback*" is to provide the same feedback to the user as to the robot. This is based on the intuition that if the user shares the same perception medium as the robot, instructions to the robot will be far more reasonable and sensible. For example, in the case of a visual target to servo to, the *User* can simply select a target from the same visual data stream that is also shared by the robot.

"*Qualitative instructions*"*—How should the robot be instructed?*
Commanding a robot can be a difficult and troublesome task. To overcome this problem we prescribe the use of qualitative high-level instructions. These instructions offer or suggest information that are easily understood and are relatively natural to use. For example, instruction such as, "*Go forward*" until you can't, "*keep along*" following that wall and then "*go through*" this door.

"*Qualitative explanations*"*—How should the representation be provided to a User?*
There is a need to describe to a *User* what is happening during the course of a robot mission. This description should include what events and activities took place, in a given period of time. For example, a mission can be described as, "went forward", "went along a wall", and then "gone through a door".

"*User interface*"*—What controls should we provide to the User?*
To complete the system, consideration must be given to the development of a user-interface. It provides a means for the display of "*Instructive feedback*" and abilities for the *User* to give "*Qualitative instructions*".

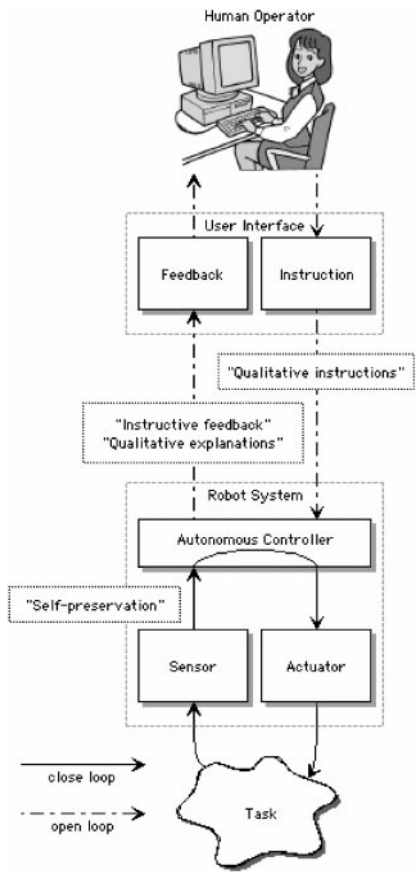


Figure 1. Conceptualization of *Supervised Autonomy* (see text for further discussion).

The general concept of our paradigm is to incorporate supervisory control with a qualitative approach for the control of robots. The overall concept of our paradigm can be seen in Fig. 1. The placement of the components described above are shown within a supervisory control framework, the solid line denoting tight/strict control, whereas the dotted-line denote loose/not-strict control (cf. Sheridan, 1992). Supervisory control does not rely on human users to perform all the basic functions of perception and action in a system (Sheridan, 1992). The approach we have taken shifts all basic autonomous functions to the physical robot agent, integrated with a set of qualitative instructions, in combination with a simple graphical user interface, and together with suitable feedback form the complete framework.

As an experimental test-bed for our approach we have chosen the task of teleoperating a mobile robot. This task is well defined problem and an excellent framework for evaluating our ideas (see Sheridan (1992) for an extensive coverage of this topic). A well integrated system should provide a user with the flexibility required to travel in complicated and dynamic environments, without requiring *a priori* knowledge of the environment.

A description of our experimental setup is provided in Section 2. In Section 3, the implemented components which encompassing the consideration taken above are presented. A presentation of the experimental result is given in Section 4. A conclusion is provided in Section 5.

1.1. Related Work—Teleoperation

Teleoperated robot systems have been a major area of interest in robotics research. Much of the research in teleoperation was initiated during the nuclear reactor era of the 1960s (Corliss, 1972). The high level of interest were due to the large number of applications that were typically performed in hazardous environments that were too dangerous for humans. A recent example of a robot operating in a hazardous environment was performed by the Dante II robot, in their exploration of the volcanic mountain, Mt. Spurr (Apostolopoulos and Bares, 1995; Wettergreen et al., 1995). Sheridan's book describes many other examples, includes toxic waste cleanup, military operations and mining (Sheridan, 1992).

Our interest in telerobotics is focused on providing an autonomous navigation system for mobile robots that can accept high level instructions from human operators. Our work is inspired by the problems of continuous close-loop control that are associated with most teleoperational systems. Typically, such systems are of the direct control style that require constant monitoring by an operator of the robot's actions while the operator is sending a stream of commands to the robot. A side-effect of constant monitoring the robot's state and controlling the robot by the operator is the resulting burden of not just the overall mission but also the general safety of the robot. This produces an effect that (Arkin and Ali, 1994) refer to as "cognitive overload". We share the same belief with Arkin and Ali that by removing the burden of constant commanding and monitoring helps to reduce the operator's cognitive load.

Some common strategies for relieving this load include providing a set of pre-programmed routines (Corliss, 1972). Other approaches use "planning by trying it out on a computer simulation first" (Sheridan, 1992). Most of the reported work in simulation relies strongly on the use of environmental models that must be sufficiently well understood before any pre-set routines can be derived and programmed. By taking the simulation first strategy, the solution to the problem relies completely on the accuracy of the modelling of the real world. Such strategies lack the ability to attend to unforeseen situations that occur in real dynamic worlds. This may be possible for a robot manipulator working in a desktop environment. In the context of mobile robot situations, this is an unrealistic assumption.

An interesting approach was recently reported in the Rocky 7 Mars Rover research project. The robot Rocky 7 performs its navigation by traversing a path of way-points selected from a 2D image by a human operator (Volpe et al., 1996). This approach removes the need for a complex path planner and allows the robot to use a simple algorithm to control movement between the way-points. However, while this approach is suitable for the Martian environment it can not work in dynamic worlds. For example, after the operator has selected a way-point for the robot to move towards, a dynamic obstacle could move to block the direct path or the way-point itself could move!

2. Our System

An overview of our system is depicted in Fig. 2. The full configuration of our system involves four sub-systems: a Yamabico robot (Yuta et al., 1991), a vision processor (Inoue et al., 1993), a communication server (Jung and Stanley, 1995) and an user interface console. The basic operation of our system is to control the robot over a radio link via a SUN workstation, acting as a communication server. Due to computation limitations the vision processing resides off-board, the visual data perceived by the robot is sent as video signal to the vision processor over a video transmitter. Aside from these special communication medium, the vision processor, communication server and the console all communicate through a standard ethernet network. The user console is a touch screen with mouse like inputs, providing a graphical user interface. This same configuration has been used throughout our research (Cheng and Zelinsky, 1996, 1998a; Jung et al., 1997).

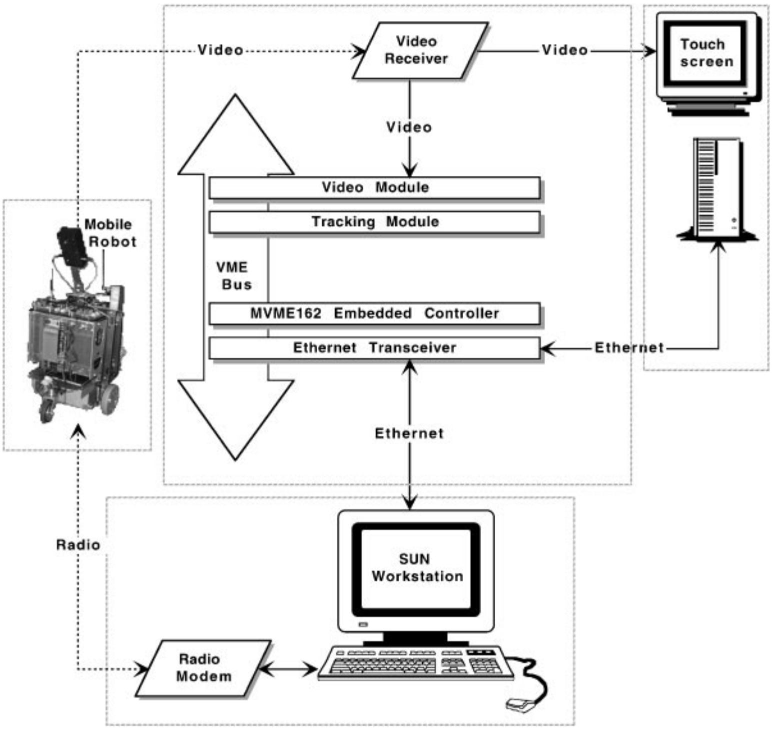


Figure 2. System configuration.

2.1. Yamabico Mobile Robot

The Yamabico mobile robot shown in Fig. 3 was developed at Tsukuba University (Yuta et al., 1991). The robot is designed to be a small self-contained autonomous robot, it has a multi-processor based architecture. All of the processor modules communicate through the Yamabico-bus, using a Dual-Port-Memory mechanism. The robot has a MC68000-CPU master module, running the MOSRA OS (Yuta et al., 1991). MOSRA provides the following features: process management, hardware interrupt handling, exception handling, memory management and interprocess communication. A transputer-based (T-805) locomotion module provides all of the motor feedback and control of the robot (Iida and Yuta, 1991). This locomotion module is designed to follow a given trajectory, using feedback information from the robot's wheel encoders. The locomotion module operates as a digital PID controller to govern motion of the robot. An ultrasonic module is also provided on the robot, but it is not used in our experiments. In addition, a radio modem, a small size CCD camera and a video transmitter have been included in our system. The modem has a maximum bandwidth of 9600 bps. It is used to obtain the results from the vision processor via the communication server. The video transmitter and camera provide video input to the vision system. The video transmitter has a maximum broadcast rating of 1 km.

2.2. Communication Server

The communication server is a SUN-workstation (running Solaris OS) with a radio modem attached. The communication server was established for a number of reasons: it allows single point communication to the robot from anywhere on the ethernet network without needing to be inside the range of the radio modem; a more powerful machine can be used to provide the management if required; and it removes the restriction of having resource demanding routines residing on the robot, as these routines can be executed elsewhere.

All data communication is done using a layer-based communication software developed by Jung and Stanley (1997), named *Radnet*. *Radnet* provides all data packet routing to the robot from the vision system via the communications server. The server manages all network traffic between the vision system and our mobile robot. For example, once a image frame is processed,

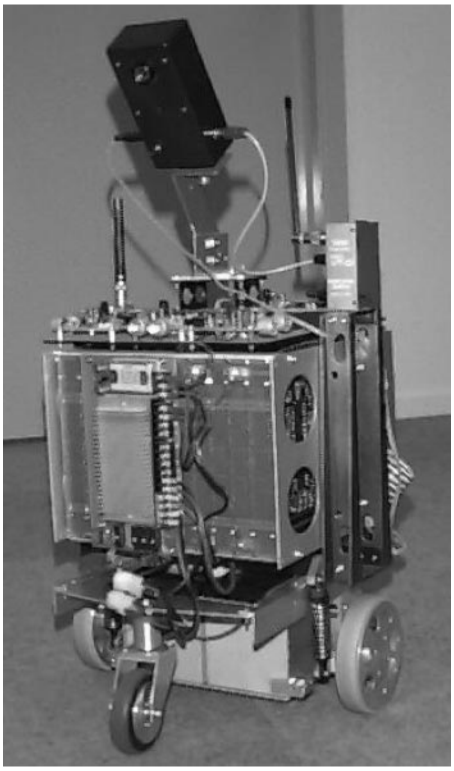


Figure 3. Yamabico mobile robot.

a command from the vision system is then sent back to the robot, guiding it on its way.

2.3. Vision Processor

The vision processor used in this research was originally developed by University of Tokyo (Inoue et al., 1992), and manufactured by Fujitsu Co., Japan. The processor is designed for template-based tracking via image correlation of a live video steam. The correlation is performed using a onboard video processing chip, which uses a technique called "Sum of Absolute Differences" (**SAD**), the basic formulation is given by Eq. (1). The system comes with two processing modules, a video module and a tracking module. The system is designed to support up to five tracking modules simultaneously. Currently, only one tracking module is used. The modules are connected via a special vision-bus—a ribbon cable connected to the front of each module. The modules are powered via a VME-bus backplane, and a Motorola MVME-162 Embedded Controller running the VxWorks® operating system.

Figure 4 shows a screen dump of matching a set of templates of clear floor with a live video frame

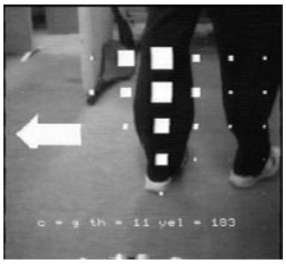


Figure 4. Template matching.

containing obstacles. The correlation values are shown in the figure as white squares. The size of the squares represent the magnitude of the correlation values. The better the templates match to clear floor the smaller the squares that are displayed.

$$D = \sum_{x}^{m} \sum_{y}^{n} |w(x - m, y - n) - f(x, y)| \qquad (1)$$

where m and n are the *horizontal* and *vertical* size of the template, $w(x, y)$ is the greyscale pixel value of the template at the specified coordinates, $f(x, y)$ is the greyscale pixel value of the live image, D is the correlation value between reference image w and live image f.

2.3.1. Lighting Adaptation. As an additional measure, we have incorporated a real-time lighting adaptation scheme to the basic vision processing. This scheme was first introduced and used for the tasks of the soccer playing robot and goal-oriented navigation (Cheng and Zelinsky, 1998a, 1998b). Our adaptation scheme exploits the existing correlation method used in the vision processor. This scheme allows our system to perform efficiently and robustly under various lighting conditions.

This adaptation is based on altering the templates to be matched in the next frame, using the average intensity of the current matching region in an image (see Cheng and Zelinsky, 1998a, 1998b). An example of this adaptation is shown in Fig. 5. The size of the black squares indicate the level of correlation, after adaptation the correlation improves markedly.

3. Components

We have embraced a behavior-based architecture and our system has been built with a set of high performance real-time vision-based behaviours (Cheng and

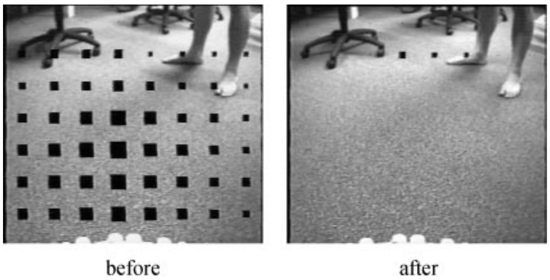


Figure 5. Adapting to lighting changes.

Zelinsky, 1996), integrated with a set of qualitative instructions. High-level feedback is provided by the use of Purposive Maps (PM) (Zelinsky and Kuniyoshi, 1996). In this section will outline the components that we have developed for our system.

3.1. Basic Behaviors

Our initial focus has centered on the development of a set of behaviors that perform basic mobile robot navigation. Throughout our research, we have progressively built a collection of vision-based behaviors. The sophistication of our system was increased by combining the various behaviors. A set of "*Basic behaviors*" have been used throughout our system, *Collision Avoidance*, *Free-Space-Move* and *Goal-Seeking*. A coverage of these basic behaviors can be find in Sections 3.1.1, 3.1.2 and 3.1.3. Their function and their usage in this framework are discussed in the subsequent sections. Moreover, they have been designed to allow easy customization for other purposes. For example, for landmark-based navigation, vision-based vacuum cleaning and soccer playing (see Cheng and Zelinsky, 1996, 1998b; Jung et al., 1997, respectively). Due to this flexibility and generality of these basic behaviors, we recognised that they have been chosen to support the development of our framework.

Some of the key attribute of these basic behaviors can be summarized as follows:

- the vision processing supporting those behaviors exhibit real-time performance, at video rate (30 Hz).
- environmental changes has been carefully considered: handle lighting changes, dynamic situations (environment need not be static, such as moving obstacle is handled).
- goal-based behavior was also provided: goal-seeking, visual target servoing, pursuing and foraging.

3.1.1. Collision Avoidance. The *Collision Avoidance* behavior acts as a safeguard to protect the robot during motion. The modular computational structure of the Collision Avoidance behavior can be simplified into *Detection* and *Avoidance* schemes. The detection phase involves determining the availability of free-space ahead of the robot for safe motion. If insufficient free space is available, the robot suppresses its other behaviours and activates the avoidance scheme. In general, the robot is set to spin in its place. Commonly, a situation in which this behavior can be activated is when the robot is not able to move away from obstacles, both static and dynamic, such as people and other robots. Also, this behavior acts as a backup to the *Free-Space-Move* behavior, which is explained in the next section. Figure 6 shows an instance of the *Collision Avoidance* behavior during a navigation experiment. The figure shows the robot wandering through the free space of the laboratory, after reaching a confined corner, the robot spins out of the corner and then reenters the free space areas.

3.1.2. Free-Space-Move. *Free-Space-Move* provides the robot with the ability to move freely within its environment. In effect, this behavior provides obstacle avoidance in static and dynamic situations. This behaviour utilizes a vision processing of searching for

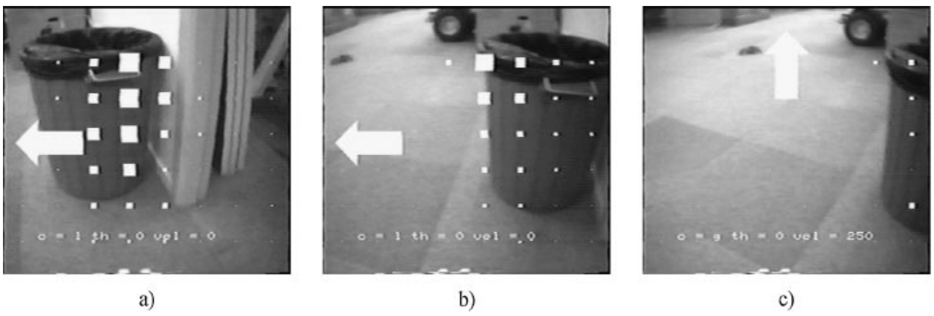


Figure 6. Collision avoidance.

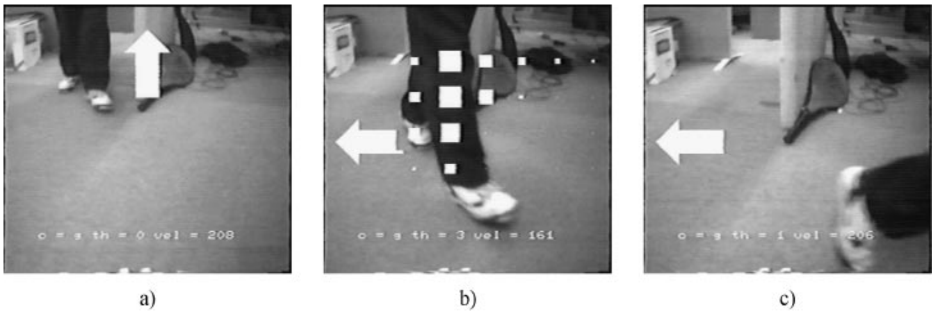


Figure 7. Free-space-move.

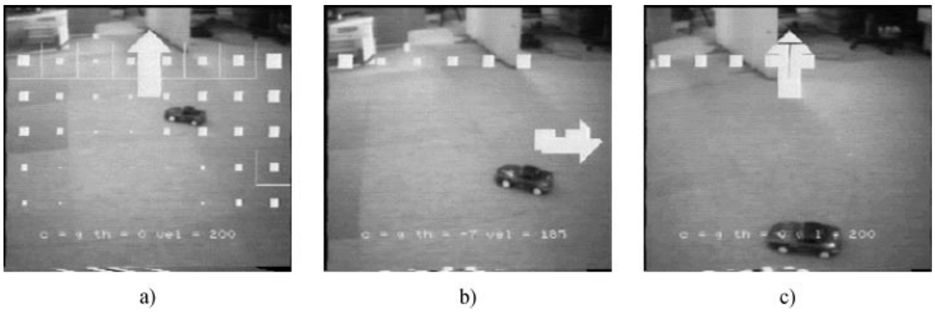


Figure 8. Goal seeking.

free space to determine the best possible motion for the robot to moves. Figure 7 shows an instance of our robot moving away from moving person while travelling forward.

3.1.3. Goal-Seeking. Once a goal has been detected, the robot executes the associated action for that particular goal, for example, visual servoing. Figure 8 shows an example of the *Goal-Seeking* behavior at work. In this experiment our robot visually servoes a target. This behavior is being utilized in the *Qualitative instructions*, for example, "Go Toward", see Section 3.5.2.

3.2. *Purposive Map*

In our system we use Purposive Maps (PM) to provide a representation for navigation (Zelinsky and Kuniyoshi, 1996). A navigation control structure which encompasses a topological like representation, removing the essence of geometric spatial detail of an environment. The basic structure is represented by connections of landmarks, for each landmark an action is associated (for example, turn left 45 degrees). Connecting these landmarks are behaviors and utilities which are required to reach the next (for example, "Contour follow" for 400 mm, or just "Go Forward"). Navigation is performed via an arbiter, coordinating and monitoring behaviors with the representation provided by the PM. For implementation purposes, a PM is simply represented as a sequential list data structure, connecting each of the behaviors as an ordered sequence in the list. Using this method for navigation we were able to yield a number of systems, Landmark-based navigation (Cheng and Zelinsky, 1996) and Autonomous Goal-oriented navigation (Cheng and Zelinsky, 1998a). The PM used to performed goal-oriented navigation was able to avoid obstacles which interfered with reaching a goal, i.e., autonomously performs obstacle avoidance.

In the original work of Zelinsky and Kuniyoshi (1996), navigation was performed with sonar and odometry. This incurred a number of drawbacks, the goal location is heavily dependent on the accuracy of odometry. Sonar information for obstacle avoidance is often erroneous and slow, limiting it to work only in a static environment. In our current work visual cues have been incorporated, removing the reliance on an accurate odometric system. Vision also increases the speed of the system, allowing it to work in both static and dynamic environments.

Due to the qualitative nature of PM and the visual information, a more intuitive and natural representation can be presented to the user. Using a PM, the robot is able to communicate to the user how well the robot is progressing in the execution of a task, and what behaviors within the robot were triggered during task execution. Our usage of a PM in providing "Qualitative explanation" is presented in Section 3.6.

3.3. Self-Preservation

For the self-preservation part of our framework, two behaviors are being utilized, *Collision avoidance* and *Free-space-move*. As suggested in the previous section these behaviors have been taken from a set of available basic behaviors. These behaviors have been implemented via two real-time visual detector, collision detector and free-space detector. The key idea behind these detector is the determination of free-space, similar idea have been proposed by (Horswill, 1994).

The self-preservation abilities provided are as following:

- Collision Avoidance—the basic prevention of collision, it is used as the general fallback mechanism for the other behaviors.
- Free-space-move—the general ability in avoiding obstacles (static or dynamic) by the determining a safe trajectory at each video frame cycle, hence, emerging a behavior which moves the robot around obstacles.

3.4. Instructive Feedback

The use of vision endorse the key notion of this component, both the user and the robot are provided with the same visual display, this can be seen in Figs. 2 and 15. In our configuration, video is provided to the user via the console (see Section 3.7 for further detail), it is also duplicated and fed to the vision processor. A feature that we have incorporated into our system to demonstrate the advantage of this aspect, is to allow the robot "Make a suggestion" to the user. This is done by allowing the robot's own capability to determine a suitable feature in the environment, the robot suggests to the user a suitable target for operation, for example in visual servoing.

3.5. Qualitative Instructions

As suggested earlier, instruction given to a robot should be easily understood and are relatively natural to use. Our motivation, in the context of navigation, has been to provided natural navigational instructions which are used throughout the everyday life of a person. For example, "Follow this wall.", "Move forward.", "Go between these two objects." and "Go to this place". The experimental results will show the usage of these instructions.

To demonstrate this part of our framework we have implemented the following simple qualitative instructions:

- Go Forward—allows user to instruct the robot to move ahead, the Self-preservation component of the system will prevent collision from occurring.
- Go Toward—allows user to select a target, in order for the robot to move toward, i.e., visual servoing.
- Go Between—allows a user to instruct the robot to travel between two objects. For example, to move between the jambs of a doorway.
- Keep To—allows the robot to keep long a boundary, typically it can be used for directing the robot to follow a wall.

3.5.1. Go Forward. This instructs the robot to start moving ahead in its current direction until the Collision Avoidance behavior described in Section 3.1 halts the robot and notifies the operator of this event. An example is shown in Fig. 9. Each notification is recorded in a PM.

3.5.2. Go Toward—Visual Servoing. The "Go Toward" qualitative instruction is used to instruct the robot to move toward a given target, that has been selected by the operator. Various effects can be achieved by using this behavior, such as visual servoing, following and posing. Visual servoing can be simply achieved through the operator selecting a static object and the robot moves toward the selected target. A side effect of visually servoing to an target, is that if the target were to move the robot will follow it. Posing can be achieved by setting the minimum distance with which a servoed target can be approached.

Figure 10 shows an example of the "Go Toward" instruction. In this case the user selected the toy car as a landmark. The robot servos on this target. If the car moves as shown in the example, the robot follows it

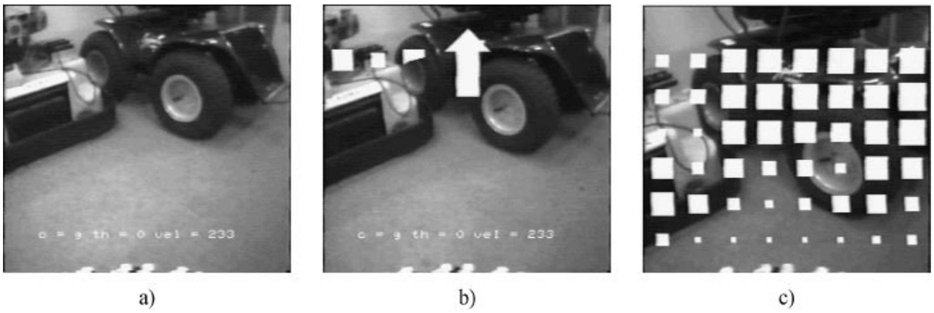


Figure 9. Go Forward—Qualitative behavior.

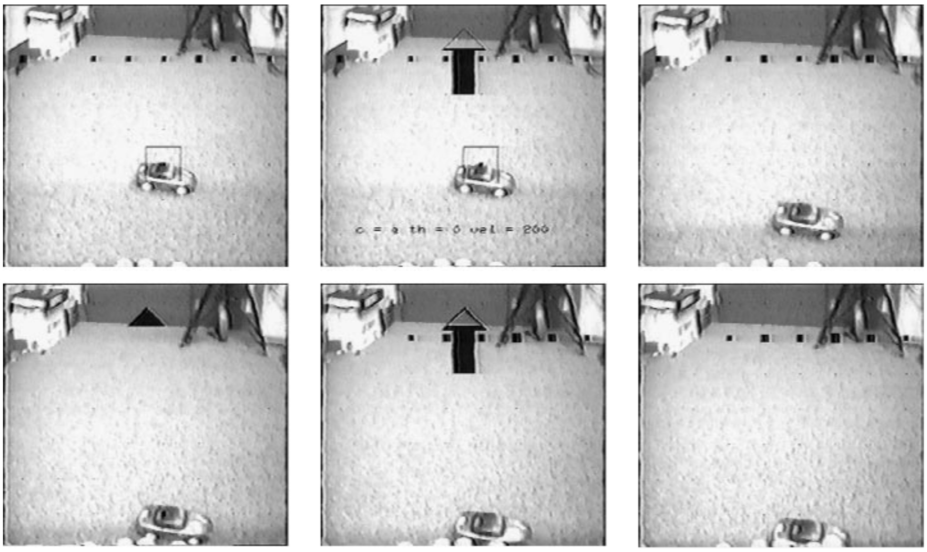


Figure 10. Go Toward—Qualitative behavior.

autonomously. If the robot loses sight of the landmark, the robot stops and notifies the operator.

3.5.3. Go Between. The "Go Between" instruction allows the operator to select two horizontal template images that are used as reference landmarks. The robot is instructed to travel between the two reference landmarks. The motion trajectory is calculated by determining the average floor position of these two landmarks. Figure 11 shows the robot being instructed to pass through a doorway using the "Go Between" command.

While traveling between the two reference landmarks the vision system tracks the two image's positions. If the vision system loses track of any one of these images, or if the positioning error of the images become significantly large, a notification is issued back to the operator and the robot stops. This behavior is useful for commanding the robot to move through tight spaces such as doorways, or narrow corridors.

$$x_t = x_1 + \frac{x_2 - x_1}{2} \tag{2}$$

$$y_t = y_1 + \frac{y_2 - y_1}{2} \tag{3}$$

where x_1 and y_1 are the x and y coordinates of the first template, x_2 and y_2 are the x and y coordinates of the second template.

The robot's motion trajectory is determined by the steering to the position between the two landmarks. This position is calculated using Eqs. (2) and (3).

3.5.4. Keep To. The "Keep To" qualitative instruction allows the operator to specify two diagonally adjacent

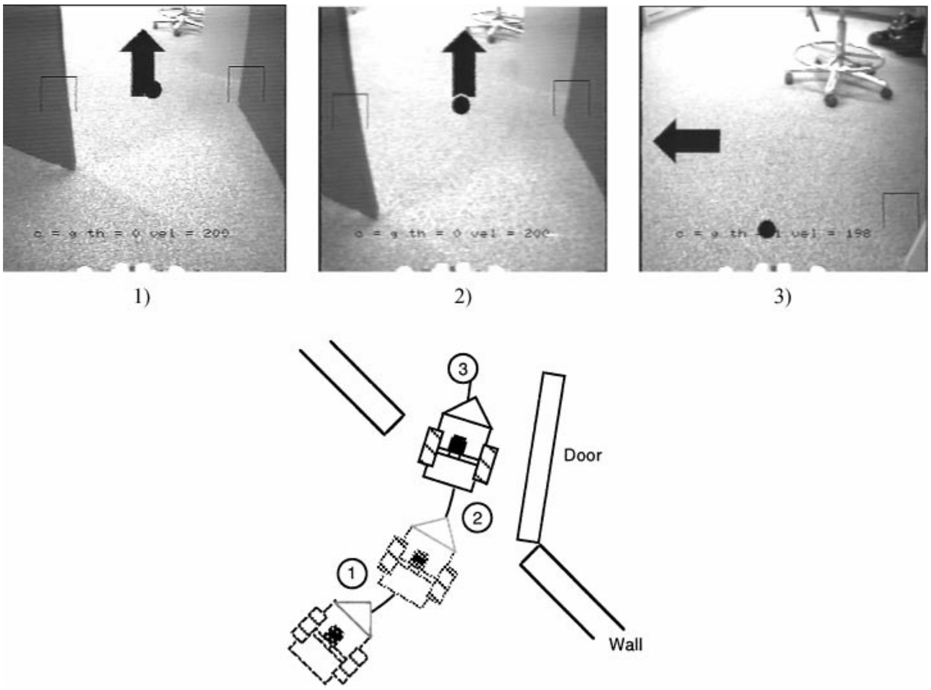


Figure 11. Go Between—Qualitative behavior.

template images. This selection determines the trajectory along which the robot will travel. For example, if the skirting of a wall is used, the robot follows the wall until it loses sight of the wall or if tracking of the reference templates fails. The calculation of the robot's trajectory corresponds to the angular relationship between the two diagonal templates. Figure 12 shows an example of the "Keep To" behavior.

$$\theta = \tan^{-1}\left(\frac{x_2 - x_1}{y_2 - y_1}\right) \tag{4}$$

where x_1 and y_1 are the x and y coordinate of the first template in the image, x_2 and y_2 are the x and y coordinate of the second template in the image.

Using the first template, which is the physically closest to the robot, the robot servos until it is a distance d from the template. This stage of processing is similar to the "Go Toward" behavior. Once the robot has reached the desired distance from the temporary goal, then the robot's alignment is calculated using Eq. (4). The angle θ determines the zero alignment for the robot to track the reference template.

3.5.5. Making a Suggestion. An additional feature that we have incorporated into our system is to allow the robot to suggest to the operator a landmark(s) that are interesting. This command instructs the vision system to determine from its current view a distinctive feature that it perceives to be a good candidate as a landmark for tracking purposes. Such a command is important because the perception capabilities of the operator are far more sophisticated than those of the robot. Therefore by allowing the robot's own sensors to make the appropriate suggestions ensures that the most effective landmarks are selected. An *Interest-Operator* has been used to identify potential goal targets. Figure 13 shows the effect of the Interest-operator. The unknown toy car is easily identified as an interesting landmark feature. Once the landmark has been located visual servoing can be performed using this distinctive feature, for example, by using the "Go Toward" instruction.

3.6. Qualitative Explanation

Explanations provided to humans should not be overly complex, they should be provided so that only the necessary information is presented. This gives rise to a mechanism that can maintain a record of events that occurred during mission execution. We use Purposive Maps (PM) to maintain a log of events, see Section 3.2.

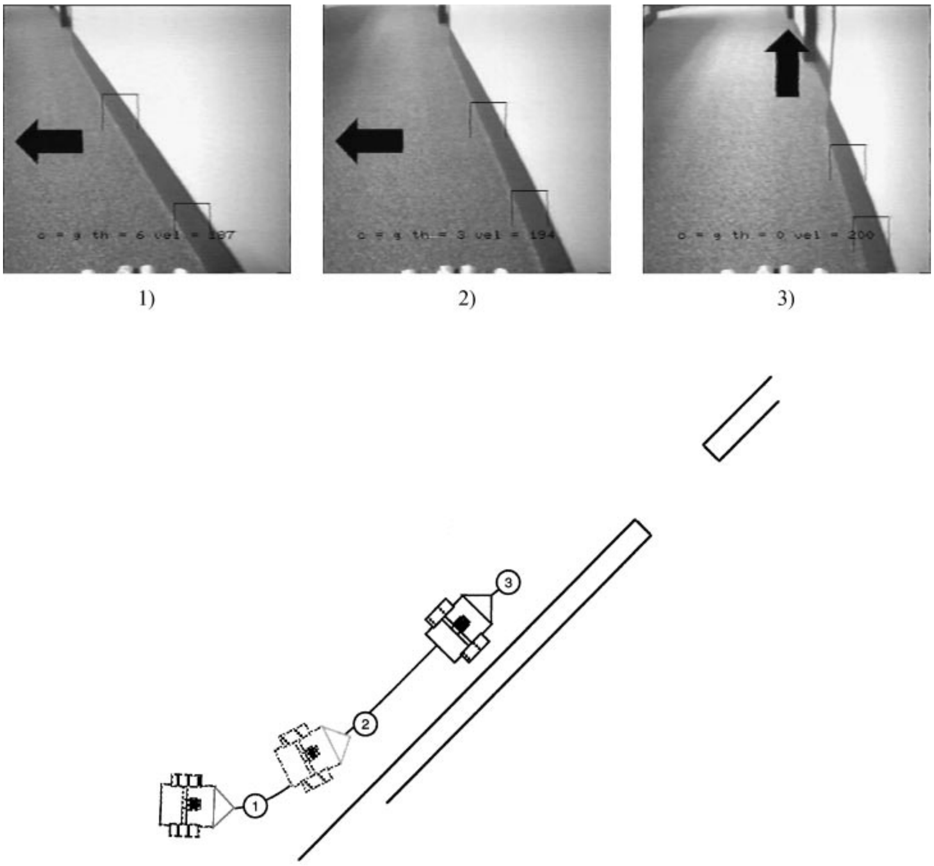


Figure 12. Keep To.

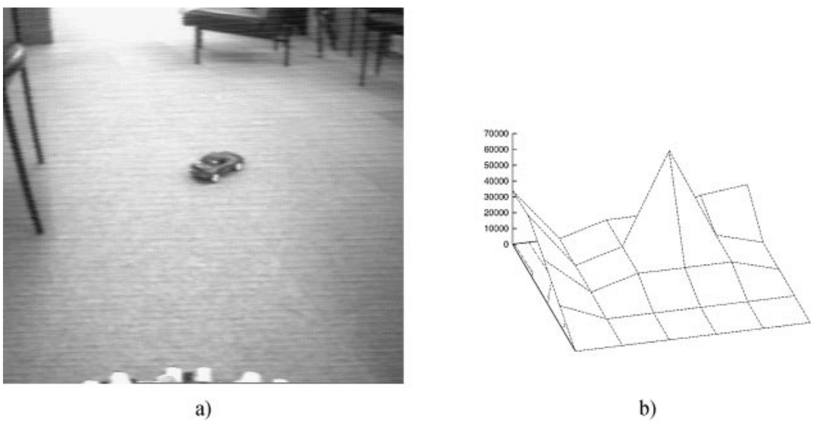


Figure 13. Making a suggestion.

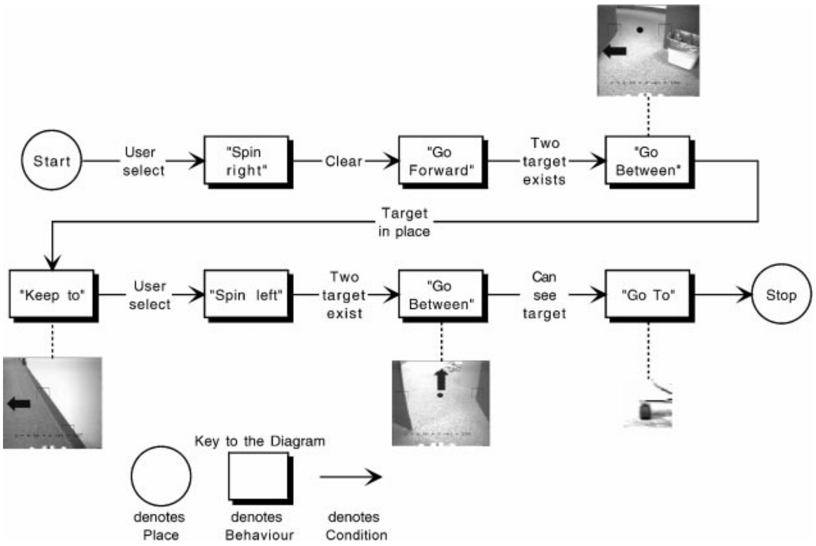


Figure 14. PM of mission.

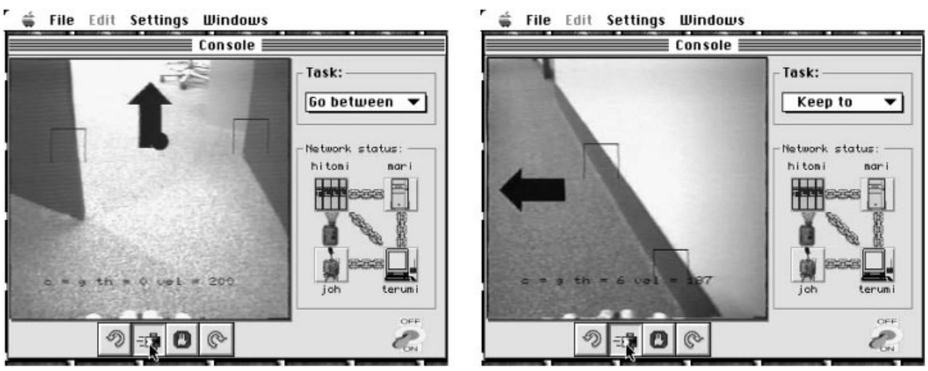
The PM provides a *qualitative explanation* of the events that occur during the execution of high level *qualitative instructions*. Thus, also removing a system's reliance on direct feedback.

A key property of PMs is the ability to express events and locations qualitatively. Other advantages of using PMs are that a general representation can be learnt from an execution of a mission. Later, the mission can be replayed. This feature further reduces the "cognitive overload" of the human user.

Figure 14 shows a "Qualitative explanation" of an experiment which was conducted using our system. Further detail of the experiment is presented below, see Section 4. The key idea of incorporating a PM, is that it can be used to express only the necessary information. The explanation showed a sequence set of behaviors, executing in a qualitative manner, providing a natural way in which information can be shared with both human and robot. In this explanation, it is easy to see from this information that, the robot at first spun right, went forward, traveled between a door, then went along a wall, turned left, went through another door and servoed toward a landmark on the floor.

3.7. *User Interface*

Figure 15 shows the graphical user interface provided to the user as a console, the prime focus of this design was to keep features to a minimum, thus producing a simple but effective interface. Physically, the interface



1) "Go Between" 2) "Keep To"

Figure 15. User console: demonstrating the **Qualitative instructions**, "Go Between" and "Keep To".

provides a touch screen, acting as a mouse, which the User selects a target by pointing at the screen. Other functionalities included: link status between of the subsystem is also provided, indicating their connection; a list of qualitative instructions is provided via a menu, allowing selection to be made easily.

We will briefly explain the main controls:

 This icon instructs the robot to begin executing a set of qualitative instructions.

 This icon commands the robot to abort the current actions it is executing.

 This icon commands the robot to start spinning right.

 This icon commands the robot to start spinning left.

The spin icons are used to give the operator a pan view of the scene when deciding which visual behavior to select for navigation from the Qualitative instructions menu.

The touch screen provides all of the functionality that can be accessed by a visual mouse interface. Thus, an operator may instruct the robot to "STOP" simply by placing his/her finger over the stop button.

Qualitative instructions are constructed by selecting from a list of menu items that are located on the right side of the main console. This ensures only one instruction is executed at a time. Beneath the menu is a status indicator which give a report about the communication status between each server. The user selects a qualitative instruction from the touch-screen menu and then selects the visual landmarks that are to be associated with this instruction, by pointing at the relevant parts in the displayed image. For example, after selecting the Go Toward instruction, the user selects a landmark in the image to move towards. In the case of the "Keep To" and "Go Between" instructions, the user selects two landmarks by touch the screen at the appropriate places in the image, as shown in Fig. 15.

Other researchers have also implemented similar user interfaces for various command systems. (Sekimoto et al., 1996), uses a touch panel to control a robot's motion to a target position. The robot has no autonomy and would drive into a wall if commanded to do so. The system that reported by (Shibata et al., 1994) also uses a touch screen for selecting an image of an object to construct a "Go toward" instruction. However, this system would also collide with obstacles if commanded to do so. These robots do not possess autonomous self-preservation capabilities.

4. Experimental Results

This experiment is intended to demonstrate our approach through the development of a system that allow the teleoperation of a mobile robot. To show how a mission can be performed using only a set of simple qualitative instructions. The robotic's mission is to travel from one room to another. Neither the robot or the user were provided with any prior knowledge of the environment. The user was able to perform the task without any difficulty.

Figure 16 shows a mission in an indoor office environment(our laboratory). At ① the robot was instructed to "Spin" right, at ② to "Go Forward", at ③ to "Go Between" and out to the corridor. Next at ④ to "Keep To" the skirting boards of the wall, then ⑤ "Spin" left, ⑥ to "Go Between" and through the door to another room. And finally at ⑦ to "Go Toward" the target on the floor. Figure 17 show snapshots of this mission as the robot travels through each of these instruction.

Figure 17 shows snapshots of a mission performed using the qualitative instruction developed for our system, described above. The basic mission was performed through only a handful of commands given by the user via a console. Only seven instruction were needed, (1) Turn right, (2) Go Forward, (3) Go Between, (4) Keep To, (5) Spin Left, (6) Go Between and (7) GoTo.

Using a PM, a "Qualitative explanation" of this experiment is given by Fig. 14. As suggested in Section 3.6, the mission could be repeated with the same PM; however, it can only be done through manually returning the robot back to its initial position.

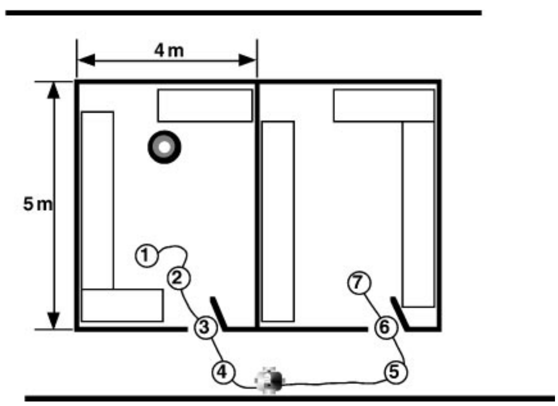


Figure 16. Mission.

Figure 17. Snapshots of a Supervised Autonomy mission: 1) Turn right, 2) Go Forward, 3) Go Between, 4) Spin Left, 6) Go Between and 7) GoTo.

5. Conclusion

In this paper we presented a developmental paradigm, *Supervised Autonomy*, in this framework a system is built in a way which can augment human facilitation. We proposed five key attributes, which interconnect into a complete human-robot interface system. The essence of our paradigm is that we emphasis the human aspects, alleviating stress on the user by shifting as much as possible of the computational and cognitive load onto the robot.

Our development can be summarized as follows:

- by shifting the general control of the robot away from the *user* back onto the robot, thus allowing "*Self-preservation*" of the robot, alleviating the burden placed on a *user*.
- by selecting suitable feedback based on the intuition, that if the user shares the same perception medium as the robot, instructions to the robot will be reasonable and sensible.
- providing a set of simple instructions for the robot, in so that they are qualitatively and naturally derived, allowing a sense of ease of use.
- provided explanations to a *User* in a natural manner, in which described to a *User* what events and activities took place.
- presented a "*User interface*" which encompassed the above requirements.

We believed that the real-time visual behaviors play a significant role in providing the autonomy to our system, which served as the foundation of the complete robot system. By adopting these components in a qualitative manner allowed us to produced a useful human-robot system.

Acknowledgments

The research was supported by a large ARC grant, provided by the Australian Research Council.

References

Apostolopoulos, D. and Bares, J. 1995. Configuration of a robust rappelling robot. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Arkin, R.C. and Ali, K.S. 1994. Integration of reactive and telerobotic control in multi-agent robotics systems. In *Proceedings of the Third International Conference on Simulation of Adaptive Behavior (SAB94) [From Animals to Animats]*. Brighton, UK, pp. 473–478.

Cheng, G. and Zelinsky, A. 1996. Real-time visual behaviours for navigating a mobile robot. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 973–980.

Cheng, G. and Zelinsky, A. 1998a. Goal-oriented behaviour-based visual navigation. In *Proceedings of IEEE International Conference on Robotics and Automation*, Leuven, Belgium, pp. 3431–3436.

Cheng, G. and Zelinsky, A. 1998b. *Real-Time Vision Processing for a Soccer Playing Mobile Robot*, pp. 135–146. Vol. 1 of (Kitano, 1998). Extended version of IJCAI'97 Workshop paper.

Corliss, W.R. 1972. *Teleoperators: Man's Machine Partners*, United States Atomic Energy Commission, Office of Information Services.

Horswill, I. 1994. Visual collision avoidance by segmentation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Munich, Germany, pp. 902–909.

Iida, S. and Yuta, S. 1991. Vehicle command system and trajectory control for autonomous mobile robots. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, Osaka, Japan, pp. 212–217.

Inoue, H., Inaba, M., Mori, T., and Tachikawa, T. 1993. Real-time robot vision system based on correlation technology. In *Proceedings of International Symposium on Industrial Robots (ISIR)*, pp. 675–680.

Inoue, H., Tachikawa, T., and Inaba, M. 1992. Robot vision system with a correlation chip for real-time tracking, optical flow and depth map generation. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1621–1626.

Jung, D., Cheng, G., and Zelinsky, A. 1997. Robot cleaning: An application of distributed planning and real-time vision. In *Proceedings of International Conference on Field and Service Robotics (FSR'97)*, Canberra, Australia, pp. 200–207.

Jung, D. and Stanley, B. 1995–1997. Yamabico autonomous mobile robots—Reference manual. Technical Report 1.5, Department of Computer Science, University of Wollongong.

Kitano, H. (Ed.). 1998. *RoboCup-97: The First Robot World Cup Soccer Games and Conferences*, Vol. 1, Springer-Verlag: Berlin.

Scott, P.B. 1984. *The Robotics Revolution*, Basil Blackwell: Oxford.

Sekimoto, T., Tsubouchi, T., and Yuta, S. 1996. An implementation of a human interface using a touch panel and visual images for controlling a vehicle. In *Proceedings IEEE International Workshop on Robot and Human Communication*. Tsukuba, Japan.

Sheridan, T.B. 1992. *Teleorobotics, Automation, and Human Supervisory Control*, 0-262-19316-7, MIT Press: Cambridge, MA.

Shibata, T., Inaba, M., and Inoue, H. 1994. Instruction use by a vision-based mobile robot. In *Proceedings of Workshop of Integration of natural Language Processing and Vision Processing*, pp. 115–121.

Volpe, R., Balaram, J., Ohm, T., and Ivlev, R. 1996. The Rocky 7 Mars Rover prototype. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Osaka, Japan, pp. 1558–1564.

Wettergreen, D., Pangels, H., and Bares, J. 1995. Behaviour-based gait execution for the Dante II walking robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pittsburgh.

Yuta, S., Suzuki, S., and Iida, S. 1991. Implementation of a small size experimental self-contained autonomous robot. In *Proceedings*

of the 2nd Int. Symposium on Experimental Robotics, Toulouse, France, pp. 902–909.
Zelinsky, A. and Kuniyoshi, Y. 1996. Learning to coordinate behaviours in mobile robots. *Advanced Robotics* 10(2): 143–159.

Gordon Cheng is currently a Science and Technology Agency (STA) Research Fellow, working in the Humanoid Interaction Lab, Intelligent Systems Division at the Electrotechnical Laboratory (ETL), Japan. Before coming to Japan he conducted his Ph.D. study at the Department of Systems Engineering, The Australian National University, Australia. Prior to his Ph.D. study he worked as a Computer Systems Officer for the Department of Computer Science at the Univeristy of Wollongong, and has also worked as a consultant and a national systems manager for a major transport company in Australia. During the time he was at the University of Wollongong, he completed his Bachelor and Master degrees in Computer Science. His research interests include, Object-Oriented Software Development, Mobile Robot Navigation, Behaviour-based Robotics, Human-Robot Interaction and Humanoid Robotics. He is a society member of the IEEE and the ACM.

Alexander Zelinsky was born in Wollongong, Australia in 1960. He worked for BHP Information Technology as a Computer Systems Engineer for 6 years before joining the University of Wollongong, Department of Computer Science as a Lecturer in 1984. Since joining Wollongong University he has been an active researcher in the robotics field, he obtained his Ph.D. in robotics in 1991. Dr. Zelinsky spent nearly 3 years (1992–1995) working in Japan as a research scientist with Prof. Shinichi Yuta at Tsukuba University, Dr. Yasuo Kuniyoshi at the Electrotechnical Laboratory. In March 1995 he returned to the University of Wollongong, Department of Computer Science as a Senior Lecturer. In October 1996 Dr. Zelinsky joined the Australian National University, Research School of Information Science and Engineering as Head of the Robotic Systems Laboratory, where he is continuing his research into mobile robotics, co-operative multiple robots and human-robot interaction. In January 2000 Dr. Zelinsky was promoted to Professor and Head of Systems Engineering at the Australian National University. Dr. Zelinsky is a member of the IEEE Robotics and Automation Society and is President of the Australian Robotics & Automation Association.