

An Online Robot System for Projects in Robot Intelligence*

GERARD T. McKEE

The University of Reading, Department of Computer Science, Reading, Berkshire, RG6 6AY, UK.

E-mail: Gerard.McKee@Reading.ac.uk

Online robot systems can significantly increase the number of students that can participate in practical robotics and artificial intelligence projects and improve the quality of the learning environment provided by educators for these topics. We present an online educational environment based on employing toys as simple robotic devices and using vision-based sensing as the basis for guiding simple object manipulation operations. The paper describes the hardware and software environment that was developed to create the online system and reports on the use of the environment to support a task requiring students to implement a simple stop-look-act control cycle. The use of the environment by the students was monitored throughout the period of the project and is described in this paper. The student evaluation of the project is also described. In both cases the results are analysed and the main conclusions are presented. In general, the experience of the students was positive and lessons learned are being integrated into future systems and projects.

INTRODUCTION

INTERNET-BASED online robot demonstrations have given the general public access to advanced robotics technology [1–4]. One of the current challenges is to incorporate these online demonstrations into extended educational programmes involving individual and group projects. In this paper we describe a project aimed at achieving this. The project involves students undertaking an assignment in robotic control as part of a taught course in robotics and artificial intelligence. The work presented builds on a project called NETROLAB that investigated the provision of a robotics laboratory service remotely accessible via the Internet [5]. NETROLAB demonstrated remote control and viewing of robot devices, including a manipulator and a mobile robot, in a simple teleoperation scenario. Although it included a number of online demonstrations [1] the main aim of NETROLAB was to investigate the provision of these services in an educational context. The valuable insights gained in the NETROLAB project are now being carried through to a current project called TORUS and ultimately towards the development of a more general framework and body of experience in the Internet-based delivery of educational topics in robotics and artificial intelligence.

The aim of TORUS (for Toys Operated Remotely for Understanding Science) is to exploit simple toys, such as diggers, cranes and bulldozers, to create simple task scenarios [6]. Toys require a minimal cost to purchase and maintain relative to industrial robots, though they do not generally

offer the fine level of control required for advanced student projects. However, when placed in the appropriate scenario they can motivate a wide range of issues in robotics and artificial intelligence research.

The first TORUS scenario constructed was a simple three-player game incorporating teleoperated control of three toy devices, namely a digger, a tower crane and a bulldozer, and emphasizing aspects of task understanding and inter-player co-operation [6]. A logical next step was to challenge students to add some level of intelligence to one or more of the toy devices, aiming, ultimately, towards an automated game scenario. Towards this end a student project was developed aimed at automating the pickup operation of the digger in this game scenario [7]. A hardware and software environment was set up to allow the students to complete the task and to allow observation and evaluation of the activity and performance of both the environment and the students. The results are described in this paper.

THE DIGGER CONTROL PROJECT

The assignment set to the students was known as the ‘digger control’ project. The task set was to automate the pickup operation of the digger in the three-player game scenario depicted pictorially in Fig. 1. The three players in the game have responsibility, respectively, for controlling a digger, a bulldozer and a tower crane, and the task was to move a ball around the three sections of the arena in the repeated sequence: tower_crane–digger–bulldozer. Three cameras, two located on the digger and tower crane respectively and the third

* Accepted 19 February 2003.

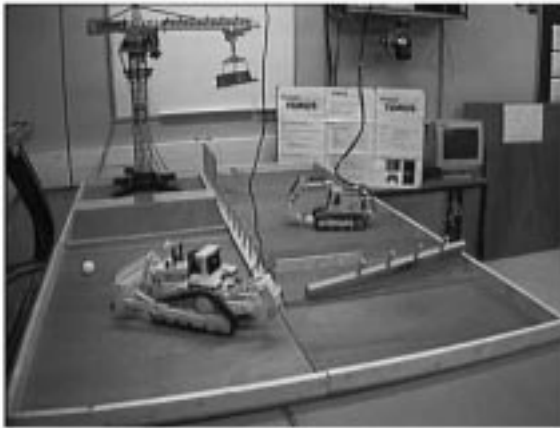


Fig. 1. The TORUS three-player 'construction-site' game scenario.

located just off the arena, provide views to the players to assist with playing the game [6].

The digger's task comprises two main elements, namely picking up the ball that is dropped into its receiving area by the tower crane, and then traversing to and depositing the ball in the drop zone, from where the ball rolls into the bulldozer's section (Fig. 2). The digger has a small camera mounted just beside its cab, providing views out into the arena ahead, including its pickup zone which is located in the area between the two forward extensions of the left and right tracked wheels. This camera provides the sole source of sensory data for the digger control project.

The digger's task can be broken out into the following six steps:

1. Locate the ball. This can be achieved with a simple scanning operation.
2. Traverse towards the ball until it is within the grasp-zone. This needs to be carried out in stages.
3. Activate the arm and bucket of the digger to affect a scooping motion to pick up the ball.
4. Locate the drop-zone for the ball. This is another scanning operation, similar to locating the ball.
5. Traverse to the drop zone.
6. Drop the ball into the drop zone.

The two basic elements of the digger task, represented by steps 1–3 and 4–6, are pickup and

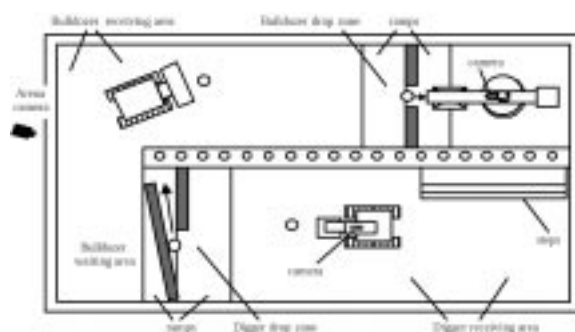


Fig. 2. The digger's task in the context of the game scenario.

Table 1. Digger task components

| Task | Localisation | Traversal | Manipulation |
|---------|------------------|--------------------------------|---------------|
| Pickup | locate ball | move to ball (grasp-zone) | scoop up ball |
| Deliver | locate drop-zone | move to & engage the drop-zone | drop the ball |

delivery of the ball respectively. These share features in common as illustrated in Table 1. Each comprises an element of localisation, traversal and manipulation, all of which are important elements of most robot tasks. The manipulation component is the simplest, since it involves open-loop operation of the arm and bucket on the digger. It is also, therefore, more prone to failure. The localisation and traversal components, however, require vision-based sensing. The traversal component can in turn be broken down into segments comprising a short traversal followed by relocalisation and fine orienting to the target (the ball or the drop-zone). Since both subtasks share elements in common it was decided for the purpose of the student project to focus on the simpler of the two, the pickup operation.

It is possible to develop solutions for each of the task components at a number of levels of sophistication, and indeed the environment could therefore be exploited in robotics and artificial intelligence teaching at a number of levels, for example at pre-university, undergraduate and graduate levels. The students involved in the current project were being introduced to robotics and artificial intelligence, and therefore a minimal level of sophistication was expected. The general requirements on the students were to understand the organisation of the task as a 'robotic' task, to understand the role played by the vision data provided by the camera in performing the task, and to understand the way in which this can be linked with the controls to create a stop-look-act control cycle. Ultimately the students were to implement and test such a stop-look-act control strategy using the digger as a real robotic device.

An important component of the task was to perform a number of measurements on the image data provided by the camera. All of the tests require locating the ball in the image. The first, during the localise stage, requires detecting just the presence of the ball in the image. The second, during alignment, requires determining whether the centre of the ball is within some threshold distance from the centre of the image. The third, during traversal, involves determining whether the ball is in the grasp zone. The students were given a short introduction to histogram-based image segmentation techniques as a suggested method for carrying out these measurements.

In order to ease the process of marking the projects the students were required to define labelled buttons in their application interface,

labelled scan, align and traverse, that initiated each element of the task separately, and one labelled pickup that caused the full sequence of steps to be executed automatically. The student's interface was to include the image downloaded from the camera and useful additional features including display of the image pixel histogram and a segmented image. They were also required to implement a set of manual controls to provide for manual manoeuvring of the digger and manual selection of the location and size of the ball in the image. The students were advised that the manual controls provided a means of 'advising' or 'overriding' the parameters of the automated system and a fallback position for students who could not successfully get to grips with the image processing functions. This meant that all students should in principle develop a 'complete' working system incorporating 'shared' control.

THE ONLINE ROBOT SYSTEM

The hardware and software for the project were largely drawn from existing software that had been developed for the TORUS construction site game scenario. The overall system configuration is illustrated in Fig. 3. The main hardware element was the interface to the digger. The toy digger (from Funrise Inc.) comes with a wired joystick pendant for controlling the two tracked wheels on the digger, the motion of the digger arm (up and down) and the motion of the bucket on the end of the arm (in and out). The arm and bucket can be moved up-down or in-out within physical limits defined by the body of the digger. A miniature camera was mounted in the belly of the digger, providing a view of the area towards the front of the digger. When the arm is down the view towards the horizon is obscured. In order to wire the toy into the computer the wired connection to the control pendant is cut close to the pendant. Male and female 9-pin D-type connectors are inserted into the pendant cable to allow for computer

connections and to retain the pendant for local manual control of the digger. Computer control is by means of an 8-relay ISA bus card. The relays are wired in pairs for two-way control of each of the four motors on the digger. The PC on which the card is mounted runs the Linux operating system. A loadable Linux module was created to provide the software interface to the relay card.

The main software comprises two servers, a control server for controlling the digger and the video server for grabbing images from the camera mounted on the digger. Both servers were implemented using the C programming language and were multi-threaded—a separate thread is created for each user. Users have to provide a username and password to login to both servers. This provided the basis for monitoring server usage over the duration of the project on a per student basis. The control server allows the user to issue forward, reverse, arm up and down, and bucket in and out commands to the digger. The video server supports a 'grab' function that allows the capture and return of an image from a miniature colour CCD camera mounted on the digger. The video grabber card is a Hauppauge WinTV card supported by the Video4Linux module. The video server employs a separate thread to grab the image and save it to a data buffer where it can be shipped out to a client. It supports grayscale and RGB24 image formats at a range of resolutions (default 352×288 pixels). Since the digger was restricted to a small, enclosed area, the video, control and power signals were all delivered by a tether. Both servers supported a telnet-based connection. This was particularly important for students to confirm that they can login, to confirm that the servers are live, and to test out the command interfaces (commands and responses) separate from their programs. If multiple users are logged onto the control server they were queued, since only one user is allowed to control the digger at a time. A time limit was set on usage, which varied between 10 and 5 minutes during the course of the project. The video server was not

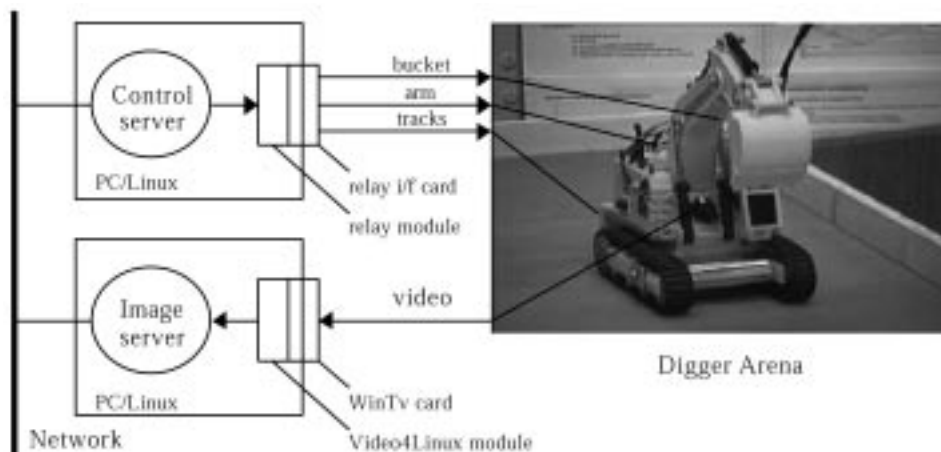


Fig. 3. Overall system architecture.



Fig. 4. Sample Delphi client application.

queue-based. The servers were running on separate PC/Linux platforms, the control server on a 486 PC and the video server on a P100 PC. Both servers were set to allow a maximum of 20 concurrent users.

The students were required to implement their programs using Delphi, an Object Pascal-based programming platform. A small library of Delphi software provided basic functions for connecting to the server, logging in, issuing commands to the control and the image server, and retrieving the grabbed image data from the image server. The latter included functions to retrieve raw images from the server in blocks of 1 Kbyte. A sample executable program demonstrating a graphic interface that included facilities for downloading and displaying an image was also provided (Fig. 4). The interface to the application represented an extension to an assignment that the students had undertaken at an earlier stage in their course. That assignment required the students to write a program to login and manually control the digger, but excluded image acquisition, display or image processing functions. All documentation, source code, and technical specifications could be downloaded or browsed from a website that was specially set up for the project.

THE ASSIGNMENT: RESULTS AND EVALUATION

The project assignment was set to a body of just under 100 students who had completed an introductory programming course using Delphi. The assignment was set in a module on topics in

artificial intelligence and robotics. The students were given six weeks in which to complete the project.

In general the project progressed as expected of a student assignment: some students got started early and used the environment extensively, the majority started work in earnest within two to three weeks of the deadline, and still others started work on the project in the last week. The majority of students completed the assignment and a small number excelled. Many students delivered interfaces following the style of the sample application, but with additional facilities for displaying a histogram and a thresholded image, and for toggling between manual and automated control of the digger. The students employed a wide range of techniques to locate the ball, most based on the suggested histogram-based method. Some methods were very crude, for example assuming that the region of the ball contained within it the brightest pixel in the image, while others involved more advanced region-growing and boundary detection techniques.

Figure 5 shows an example of images retrieved from the server under natural room (strip) lighting.



Fig. 5. Typical images from digger camera.

Table 2. Response to student questionnaire

| Rating | | 1 | 2 | 3 | 4 | 5 | | Total |
|--|-------|---|----|----|----|----|----------|-------|
| How did you find completing the assignment | Easy | 0 | 6 | 7 | 25 | 17 | Hard | 25 |
| How useful was the code library | A Lot | 4 | 10 | 16 | 20 | 5 | A Little | 55 |
| Was using the code library | Easy | 6 | 9 | 14 | 22 | 4 | Hard | 55 |
| Was documentation given | Bad | 3 | 14 | 15 | 15 | 8 | Good | 55 |
| Was sufficient access to the servers given | Yes | 2 | 17 | 9 | 14 | 13 | No | 55 |
| Have you learnt | A Lot | 7 | 34 | 9 | 4 | 1 | A Little | 55 |

The left image shows the ball located at 30 cm from the digger and the right shows the ball just on the threshold of the grasp zone. During the project some improvements were made to the lighting to make the ball stand out more sharply against its background. This was achieved largely by the introduction of ultraviolet lighting.

In order to evaluate the educational experience the students were asked to fill out a questionnaire that posed a set of six questions, shown in Table 2. The students were asked to rate their response to each question with a number in the range 1 to 5 according to the categories associated with the response (easy to hard, etc.). 55 students responded and the results are tabulated in Table 2.

The main conclusion from rows 1 and 6 of Table 2 is that the students found the project challenging and they also learned a lot from it. Their response is depicted in Fig. 6. These results appear to reflect the open-ended nature of the project in respect of the techniques that could be used for locating the ball in the image and in the sophistication required of the user interface. It appeared to be influenced also by the fact that

students were interacting with a ‘real’ environment that provided various challenges of its own, including for example the poor repeatability of the digger movements.

Rows 4 and 5 of Table 2 show that the students were more mixed in their assessment of the accessibility of the servers and the quality of the documentation. Finally, the students found the code library both not very useful and hard to use (rows 2 and 3). This might reflect the fact that this student group had little familiarity with library management at this stage in their education.

The logs maintained during the project provided the means for monitoring student usage of the servers on a per-student basis. The analysis of the logs is presented in Figs 7 to 9. Figure 7 shows the number of logins per day over the period of the assignment and during the seventh week when the students demonstrated their programs. The main observation is the progressively increasing usage of both servers, following a roughly linear rise during the first four weeks and then an approximately exponential rise during weeks five and six. The amount of daily usage

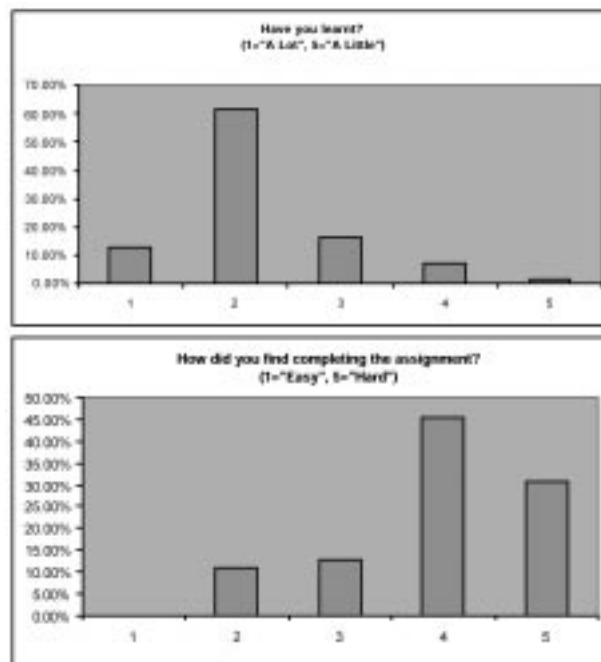


Fig. 6. Student learning and project challenge.

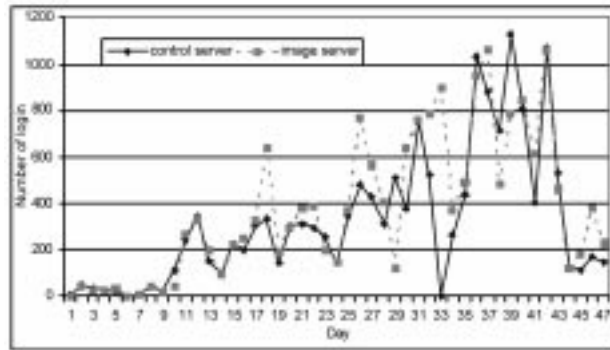


Fig. 7. Number of server logins per day.

was similar for both servers. This appeared to reflect the students' tendency to login to both servers when testing their programs, even though they were probably only testing a component of their program dealing with just one of the servers. The image server, however, did appear to take a heavier load per day than the control server, probably reflecting the students' limited prior familiarity with this server, their unfamiliarity with the image-grab functions, and the strong emphasis in this stage of the project on taking image-based measurements for locating the ball. The zero logins for Friday of week 5 (day 33) were due to the control server being down for that day.

Figure 8 shows the number of logins per student for both servers. The main observation is that some students used the servers a lot, over 400 logins to both servers, the majority used the servers between 50 and 400 logins, and some didn't login to the servers at all. Further informal analysis didn't appear to show any correlation between the numbers of times a student logged into the servers and how well they did in the assignment. This requires further study.

Finally, the load on the servers during the project was measured by recording the number of simultaneous logins for each server. Figure 9 shows the frequencies of 1, 2 or more simultaneous logins. Two observations are noteworthy. First, the pattern of usage for both servers is very similar, showing a majority of one or two simultaneous logins, and falling off exponentially for higher numbers. Second, the maximum number of

simultaneous logins was 12 on the control server, which occurred only once, and is well short of the 20-user limit that was set on both servers. This was also well short of the loading that was expected, particularly during the last full week of the project.

SUMMARY AND CONCLUSIONS

This paper has reported on the development of an online facility to support a student project aimed at motivating educational issues in robotics and artificial intelligence. The infrastructure supporting the facility and the student project used to assess its performance were presented. The results demonstrate a pattern of usage that reflects work progression normally followed by students, namely a progressive increase in activity as the project deadline approaches. The infrastructure coped well, and indeed performed better than expected, largely because the usage expected didn't match up to that observed. These results provide both a user and a system perspective on the development of online robot environments. When developing such systems it is important both to incorporate models of student working practices and to provide facilities that will sustain a consistent and reliable service for the duration of the project assignment. The results point towards recommendations both to system providers as to how to scale online robot systems to match the student load, and to student work practices in

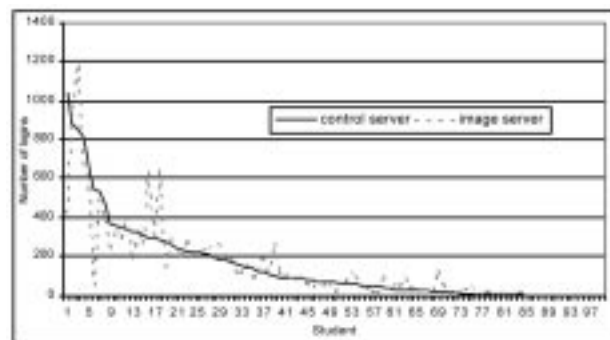


Fig. 8. Number of logins per student.

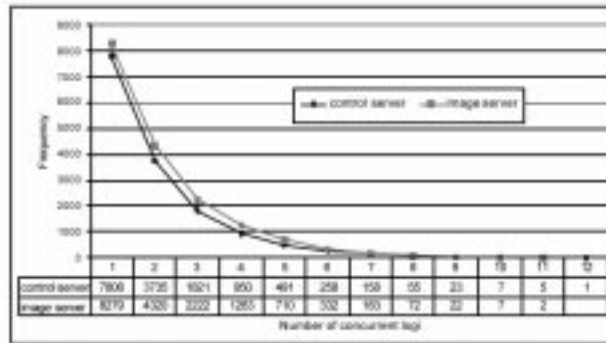


Fig. 9. Simultaneous logins.

order that they can take full benefit from the environment. Further studies are underway to develop more quantitative models and requirements.

REFERENCES

1. J. A. Fryer, Remote-control experiment using a networked robot, *SPIE Robotics and Machine Perception Technical Working Group Newsletter*, 5(1), 1996, p. 12.
2. K. Goldberg, S. Gentner, C. Sutter and J. Wiegley, The Mercury Project: a feasibility study for Internet robots, *IEEE Robotics and Automation Magazine*, Special Issue on Internet Robotics, December 1999.
3. R. Siegwart and P. Saucy, Interacting mobile robots on the Web, *Workshop on Current Challenges in Internet Robotics, IEEE International Conference on Robotics and Automation*, Detroit, MI, USA (1999).
4. M. R. Stein and K. Sutherland, Project update: sharing resources over the Internet for robotics education, *SPIE Proc. Vol. 3524, Telematipulation and Telepresence V*, 1998, pp. 180–188.
5. G. McKee and B. Brooks, Interactive robotics using the Internet, *J. Computer Science Education*, 7(2), 1996, pp. 279–290.
6. G. T. McKee and K. Phillips, TORUS: toys operated remotely for understanding science, *SPIE Proc. Vol. 4195, Mobile Robots XV and Telematipulation and Telepresence Technologies VII*, 9 pages, Nov. 2000.
7. G. T. McKee and R. M. Maunders, Exploiting toys and the Internet for robotics education, *WSES/IEEE Int. Conf. Robotics, Distance Learning and Intelligent Communication Systems (RODLICS 2001)*, Malta, September 1–6, 2001.

Gerard T McKee obtained his first degree in Electronics at the University of Manchester, Institute of Science and Technology (UMIST) in 1980. His postgraduate studies, again pursued at UMIST, focused on the modelling and design of intelligent systems. In 1987 Dr McKee joined the Computer Science department at the University of Reading, where he has taught courses in artificial intelligence and robotics. Dr McKee's research interests include robotics and artificial intelligence, with specific focus on networked robotics, modular robotics, and intelligent viewing assistance for camera control during teleoperation. Dr McKee is currently chair of SPIE's technical working group on Robotics and Machine Perception and co-chair of the subcommittee on Education Applications for the Internet and Online Robots committee of the IEEE Robotics and Automation Society.