Interacting with Computers 23 (2011) 85–95

Contents lists available at ScienceDirect

Interacting with Computers

journal homepage: www.elsevier.com/locate/intcom

"Moving to the centre": A gaze-driven remote camera control for teleoperation

Dingyun Zhu [a,b,*], Tom Gedeon [b,1], Ken Taylor [a,2]

[a] Information and Communication Technologies (ICT) Centre, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Acton, Canberra, ACT 0200, Australia
[b] School of Computer Science, College of Engineering and Computer Science, The Australian National University, Canberra, ACT 0200, Australia

ARTICLE INFO

Article history:
Received 27 January 2010
Received in revised form 28 September 2010
Accepted 18 October 2010
Available online 23 October 2010

Keywords:
Gaze tracking interfaces
Hands-busy situation
Teleoperation
Remote camera control
Rock breaking
Usability evaluation

ABSTRACT

In general, conventional control interfaces such as joysticks, switches, and wheels are predominantly used in teleoperation. However, operators normally have to control multiple complex devices simultaneously. For example, controlling a rock breaker and a remote camera at the same time in mining teleoperation. This overloads the operator's control capability of using hands, increases workload and reduces productivity.

We present a novel gaze-driven remote camera control with an implemented prototype, which follows a simple and natural design principle: "***Whatever you look at on the screen, it moves to the centre***!".

A user study of modeled hands-busy experiment has been conducted, comparing the performance of using gaze-driven control and traditional joystick control through both objective measures and subjective measures. The experimental results clearly show the gaze-driven control significantly outperformed the conventional joystick control.

1. Introduction

Teleoperation has been widely applied in a variety of situations, ranging from space exploration, inspection, robotic navigation, surveillance, underwater operations, and rescue activities. It has the promises and advantages of being able to provide replaceable surrogates for humans in hazardous or difficult working environments over long distances, potentially improving productivity and reducing costs. Regardless of whether the remote machine or robot is manually controlled by an operator, or semiautonomous, or even fully autonomous for some specific tasks, human observation, intervention, and supervision still play integral roles in these teleoperated systems (Hughes and Lewis, 2004).

Several types of user interfaces for different teleoperation settings have been characterized in Fong and Thorpe (2001), but the observations still indicate that directly controlling a robot while watching a video feed from the remote camera(s) remains the most common interaction form in teleoperation. Therefore, in most teleoperation settings, an operator's basic perceptual link to the remote environment is usually through a live video stream from a remote camera as the realistic foundation of situational awareness for the entire teleoperation activity.

In practice, operators often have to control multiple devices simultaneously to complete operational tasks, for example, controlling a mechanical robot and the motion of a remote camera at the same time. Using conventional control interfaces, such as joysticks, wheels, mouse and keyboard, will result in frequently switching hands and attention between different control interfaces. This will distract the operator from concentrating on the control task, reduce the productivity of the entire process, increase both workload and the number of avoidable operational mistakes.

In this paper, we particularly address this hands-busy problem in a situation where an operator is controlling one or more remote cameras while carrying out other teleoperation tasks. Instead of using conventional control interfaces and switching an operator's attention between tasks, we present a novel design where we use human eye gaze as an alternative input for the remote camera control using computer vision based eye-tracking technology. With the user evaluation of a modeled hands-busy experiment for an implemented prototype system, we demonstrate the effectiveness of using our gaze-driven remote camera control for resolving this common problem in teleoperation settings through both objective (performance) measures and subjective (user preference) measures.

2. Background: remote rock breaking in mining teleoperation

We consider the development of a tele-robotic control system to a giant mining equipment for rock breaking (Duff et al., 2009) as an example application of our research scenario.

As shown in Fig. 1, the rock breaker on the mine site is a serial link manipulator arm with a large hydraulic hammer at the tip to

* Corresponding author at: School of Computer Science, College of Engineering and Computer Science, The Australian National University, Canberra, ACT 0200, Australia. Tel.: +61 2 6216 7141; fax: +61 2 6216 7111.
E-mail addresses: dingyun.zhu@csiro.au (D. Zhu), tom.gedeon@anu.edu.au (T. Gedeon), ken.taylor@csiro.au (K. Taylor).
[1] Tel.: +61 2 6125 1052; fax: +61 2 6125 0010.
[2] Tel.: +61 2 6216 7151; fax: +61 2 6216 7111.

0953-5438/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.intcom.2010.10.003

break oversized rocks. The arm is installed at a Run of Mine (ROM) bin, where a number of horizontal bars (referred to as a *grizzly*) are fitted at the bottom in order to prevent oversized rocks from entering the crusher below (see Fig. 2).

The actual remote rock breaking process is shown in Fig. 3. Instead of making an operator stand next to the bin, using a line-of-sight control to manipulate the rock breaker arm, the new remote setting allows the operator to have a desktop based teleoperation environment and live videos as the visual feedback.

On the remote mine site, a number of haul trucks with ore from a nearby quarry are queueing to dump their load into the bin. The operator is required to break those oversized rocks stuck on the grizzly by operating a two-handed joystick controller. The operator has limited time to break the rocks, as trucks arrive at short intervals (about 90 s). Since dumping a load raises a large could of dust, a water spray is used to settle the dust, which requires about 30 s to make the operator have a clear vision of the bin. Therefore, the operator only has about 60 s to move the arm from its rest position, place it carefully onto a rock, break it by firing the jackhammer, and return the arm to the rest before next truck arrives.

When the operator is trying to break a rock, it is indispensable for them to have a close view (camera zoom-in view) of the target so that detailed information can be obtained to specify the spot on the rock for positioning the tip and firing the jackhammer. Otherwise, difficulties of positioning the tip on a proper spot of the rock would happen and slow the entire process, which could also result in avoidable operation mistakes. One serious issue of not having a zoom-in view in the tip positioning step is the arm would be bouncing on the rock and easily get damaged after firing the jackhammer, if there was even a small gap between the tip and the surface of the rock. We verified the need for close in zoom by discussion with an experienced rock breaker operator.

It is practically impossible to mount the remote camera on the arm to couple the camera motion to the control of the remote robot like most telerobotic or vehicle settings for reducing the control complexity, as the camera would be easily damaged when the jackhammer on the tip is being fired to break a rock. Therefore, the remote camera is actually installed on the side of the bin with a zoomed-in view transferring the live video back to the operator. The operator has to use another joystick controller to control the camera motion for adjusting the view of the target rock in order to complete the breaking spot inspection process then move on to the rock breaker arm control. This turns out to be a typical hands-busy problem that requires operators to switch hands quite often between different control interfaces.

In fact, a tip-tracking approach (Duff et al., 2009) has been introduced as a possible solution for this problem, which makes the remote camera always follow the tip by processing the position data from the sensor devices installed on the rock breaker arm and the corresponding locations around the bin. However, due to the unavoidable noise from the sensors working in the harsh mining environment, the camera motion can not precisely track the tip, especially when moving the arm from the rest position or returning it. This affects the operator to acquire insufficient visual feedback from the video stream as the remote camera may be inaccurately pointing at the spot that they expect to view.

Fig. 2. The ROM bin with a grizzly at the bottom.

In the evaluation section we will describe our experimental setting as motivated by the properties of this example real world setting.

3. Related work

Hainsworth (2001) has briefly discussed the requirements for user interfaces for teleoperation of mining vehicles and systems with the demonstrations of two teleoperated mining systems. It is clear that conventional user interfaces such as joysticks, switches, and wheels, are still the major control elements used in mining teleoperation. They are relatively simple, sophisticated, allowing teleoperation to be a viable and profitable technique, which satisfy the basic client requirements for mining systems of robustness and reliability. However, ease of use, productivity, hands-busy problem and frequently switching attention between tasks could be improved.

3.1. Remote camera control in teleoperation

Particularly for remote camera control, several alternative approaches have been developed. For example, Cohen et al. (1996)

Fig. 1. Overview of the rock breaker.

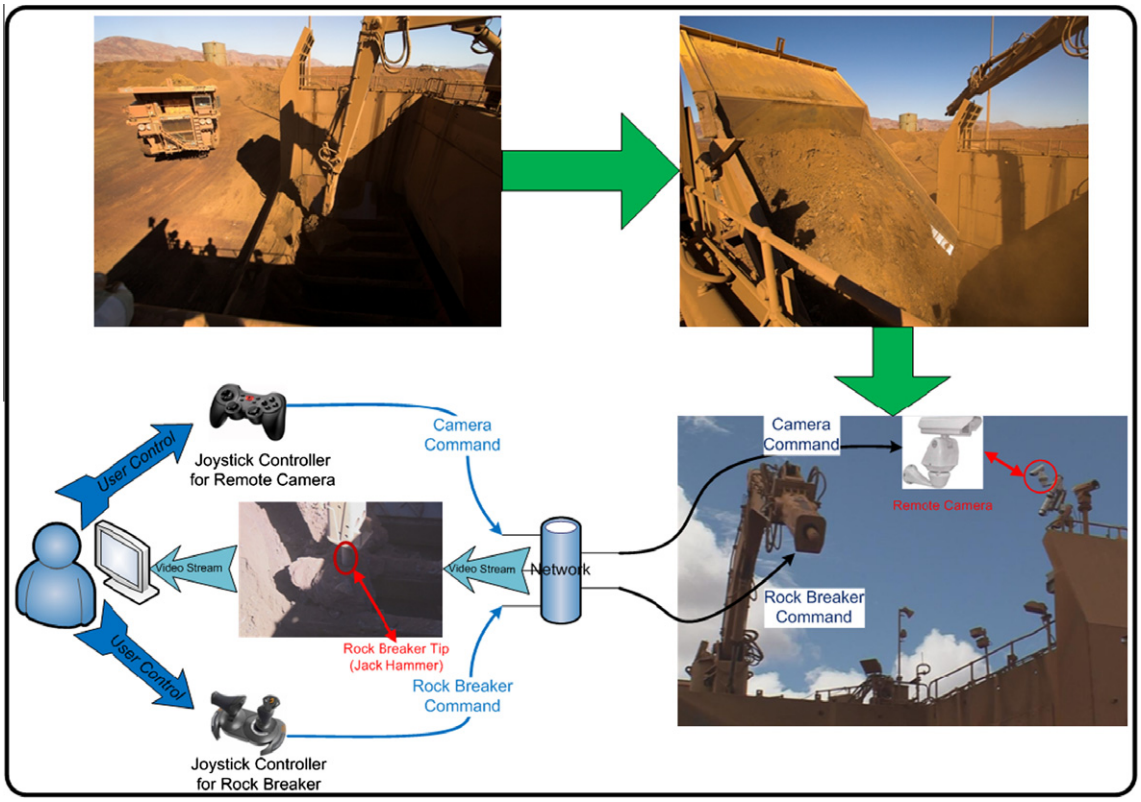

Fig. 3. Remote rock breaking.

proposed the possibility of using a set of circular oscillatory hand gestures to control a remote camera's pan and tilt motion. In addition, due to the wide popularity of the Nintendo Wii in recent years as well as its advantage of low cost, another Pan-Tilt-Zoom (PTZ) camera control system using a Wii remote and a set of infrared sensors has also been described in Goh et al. (2008). For these types of approaches, the basic intention is trying to provide more interactive or natural ways for the traditional remote camera control by alternative inputs, such as gestures. Nevertheless, they still require users to use attention and hands to operate.

In addition, head tracking has become another popular form for human computer interaction. Recent work includes exploring head tracking as an augmented input in video games in order to enhance presence, role-playing and user control (Wang et al., 2006) and controlling PTZ camera with head tracking for video chat (Yamaguchi et al., 2009). In Zhu et al. (2009), describe two different types of head tracking control techniques for the remote camera control to solve the hands-busy problem in teleoperation settings. From the results of a modeled user study where they compared the performance of using these two head tracking controls to a traditional keyboard control, they showed that head motion control was able to provide a comparable performance to the traditional keyboard control.

However, it is clear that the basic motivation of using head tracking either for virtual view point control or for real camera control is to enhance the enjoyment or engagement for the user interaction rather than productivity or performance. Moreover, compared to other alternative input modalities for remote camera control (e.g. eye tracking), head tracking requires more physical effort and may not be suitable for long-time continuous control (Zhu et al., 2010). People do not normally move their head side to side and so on in the manner which would be required for remote camera control for many hours a day. This may create occupational health and safety issues. Moving the eyes in a natural manner for many hours should not have these issues, as it is nearly fatigue-free (Saito, 1992).

3.2. Gaze-based interaction and control

As part of human's natural interaction ability, eye gaze has been recognized as an augmented input medium or control modality in "*advanced user interfaces*" (Jacob, 1991). A summary of the compelling reasons, advantages and motivations to design gaze-based user interfaces for pointing or control has been explicitly described in Zhai et al. (1999):

1. It can be an effective solution for situations that prohibit the use of the hands, for example, when the user's hands are disabled (quadriplegic) or continuously occupied with other tasks (such as the hands-busy problem in the rock breaking task).
2. Increasing the speed of user input, as clearly the eye can move more quickly in comparison to other input mediums.
3. Reducing workload, repetitive stress, fatigue (nearly fatigue-free interaction (Saito, 1992)) and potential injury caused by physically operating other devices.

Therefore, numerous approaches, techniques, applications and systems using gaze-based interaction have been proposed and developed for various situations in the field of human–computer interaction (HCI). For instance, Jacob (1991) investigated the usefulness of eye movements as a fast and auxiliary input mode with the introduction of several fundamental gaze-based interaction techniques, such as *Object Selection*, *Continuous Attribute Display*, *Moving an Object*, *Eye-Controlled Scrolling Text*, *Menu Command* and *Listener Window*. Zhai et al. (1999) presented the MAGIC pointing technique. In this approach, the cursor is automatically warped to the vicinity region of the target where the user is staring and

then they can use an additional pointing device like a mouse to manually finish the target confirmation or selection.

In addition, in order to resolve the "Midas Touch" (Jacob, 1991) problem in gaze-based interaction, Kumar et al. (2007) recently proposed a practical technique using a combination of eye gaze and keyboard triggers with a fluid look-press-look-release action, called *EyePoint*. Also, another recent approach of using modes to enable different types of mouse behavior to be emulated with gaze and by using gestures to switch between these modes, called *Snap Clutch*, has been introduced in Istance et al. (2008).

Apart from these studies on the traditional pointing and selection, there have been a variety of other attempts to integrate eye gaze into the user interface design for different interactive models. Tanriverdi and Jacob (2000) presented an interaction technique that focused on combining features of eye movements and non-command based interactions particularly in virtual environments. It uses a histogram that represents the accumulation of eye fixations on each possible target object in the VR environment, which is able to provide a profile of the user's "*recent interest*" in the various displayed objects. Similarly, there have been studies of integrating gaze-based interaction for video game control (Smith and Graham, 2006). Gedeon et al. (2008) introduced a way of using eye gaze as an alternative type of user intention for leading a group of virtual agents to accomplish cooperative tasks in a simulated game-like environment. Isokoski et al. (2007) reported another experiment on use of eye tracker with a gamepad in first person shooter (FPS) games, where they compared three control conditions: (1) a traditionally used gamepad controller, (2) the combination of gamepad controlled moving and aiming with gaze, and (3) the gamepad controller used only for moving forward and both the aiming of the weapon and steering of the movement were done by gaze. There were not significant advantages for eye operated control according to the results, but they confirmed that eye tracker input can compete in killing efficiency with gamepad input in FPS games, which could be an effective approach to minimize the use of hand controls in FPS gaming.

Furthermore, gaze-based interaction has also been used to develop specific applications for controlling real-world devices. In the late 1990s, Yanco (1998) developed a prototype robotic wheelchair system with an eye-tracking based control interface. This system allows the user to drive a wheelchair by simply looking at a set of command icons on the chair-mounted screen. Recently, Tall et al. (2009) constructed another experimental robotic vehicle which could be remotely driven by a gaze-controlled interface. In the experiment, they investigated five different control inputs (*on-screen buttons, mouse pointing, low-cost webcam eye tracker and two commercial eye-tracking systems*) for driving the robot on a racing track. From the results, they found gaze control was similar to mouse control, which provides clear evidence that robots or vehicles can be controlled "hands-free" through gaze.

We chose eye tracking as we needed a control method which could be used for many hours a day in hands-busy and attention switch settings. The control method should therefore be natural to use for many hours to avoid physical harm to the user. People naturally move their eyes all the time and eye gaze is a natural signalling technique between humans (Kobayashi and Kohshima, 2001). Whether eye gaze is suitable for this task is part of our investigation.

4. Design of gaze-driven remote camera control

In this section, we describe the design of our gaze-driven remote camera control in detail. The basic input data for this approach is the real-time raw gaze coordinate value on the screen $P_i(x_i, y_i)$. After filtering the raw gaze points into fixations, we apply the "*rate control*" mapping with a linear function gain to specify the moving angle ($CAM_{angle\_current}$) and the velocity ($CAM_{velocity\_current}$) for the remote camera to carry out corresponding pan ($CAM_{velocity\_current\_pan}$) and tilt ($CAM_{velocity\_current\_tilt}$) functions. This approach follows a simple and natural design principle: "*whatever the user looks at on the screen, it moves to the centre.*" which is similar to the "*self-centering mechanism*" in the "*rate control*".

Since human raw gaze points are inherently noisy (Yarbus, 1967), they are not suitable for direct application (Jacob, 1991). Two main forms of eye gaze are *fixations* and *saccades*. Fixations occur when a subject's eye gaze pauses over informative regions of interest and saccades represent rapid gaze movements between points. For using gaze information as a form of real-time input to control a camera, it is more suitable to use fixations as smoothed data rather than noisy raw points to avoid jerky camera movements.

We used a modified version of the *Velocity-Threshold Identification (I-VT)* fixation detection algorithm (Salvucci and Goldberg, 2000) to filter the raw gaze points from the eye tracker in real-time into fixations, as this method is straightforward to implement, runs very efficiently, and can easily run in real-time. Instead of setting a velocity threshold, a gaze movement threshold was used in the modified version, in which two gaze points separated by a *Euclidean Distance* of more than a pre-defined value are labeled as a saccade. This is because the time for receiving each gaze point is the same, therefore it is not necessary to further calculate the velocity for each point as the distance value can be directly used. In the default implementation, we chose a distance threshold of 1° of visual angle.

We can break down the entire process into the following major steps (see Fig. 4):

1. Processing the raw gaze data using I-VT algorithm to filter the noisy points (saccades), recognize the gaze fixation $\overline{P_n}(\overline{x_n}, \overline{y_n})$ by calculating the centroid of the grouped non-noisy points (Salvucci and Goldberg, 2000): $P_1(x_1, y_1), P_2(x_2, y_2), \ldots, P_n(x_n, y_n)$.

$$\begin{cases} \overline{x_n} = \frac{\sum_{i=1}^{n} x_i}{n} \\ \overline{y_n} = \frac{\sum_{i=1}^{n} y_i}{n} \end{cases} \quad (1)$$

Fixations are usually in the range of 200–400 ms (Salvucci and Goldberg, 2000), so we used 200 ms as the time interval in our default implementation, which resulted in approximately 12 gaze points per round as the eye tracker we used is able to provide a 60 Hz tracking frequency. The value of n is the number of gaze points in the fixation category after filtering out the saccade points.

2. Calculating the distance d and the angle θ between the current fixation position $\overline{P_n}(\overline{x_n}, \overline{y_n})$ and the centre of the screen $C_0(x_0, y_0)$.

$$d = |\overline{P_n}C_0| = \sqrt{(\overline{x_n} - x_0)^2 + (\overline{y_n} - y_0)^2} \quad (2)$$

$$\theta = a\tan 2(|\overline{y_n} - y_0|, |\overline{x_n} - x_0|) \quad (3)$$

3. If the current fixation $\overline{P_n}$ is in the central area C_0 (r_0 represents the radius of C_0):

$$d < r_0 \quad (4)$$

the camera will remain at the current position.
If the current fixation $\overline{P_n}$ is out of the central area C_0, the camera will start moving along the angle $CAM_{angle\_current}$ with its velocity $CAM_{velocity\_current}$:

$$\begin{cases} CAM_{angle\_current} = \theta \\ CAM_{velocity\_current} = FG \cdot CAM_{velocity\_max} \\ FG = \frac{d}{D} \end{cases} \quad (5)$$

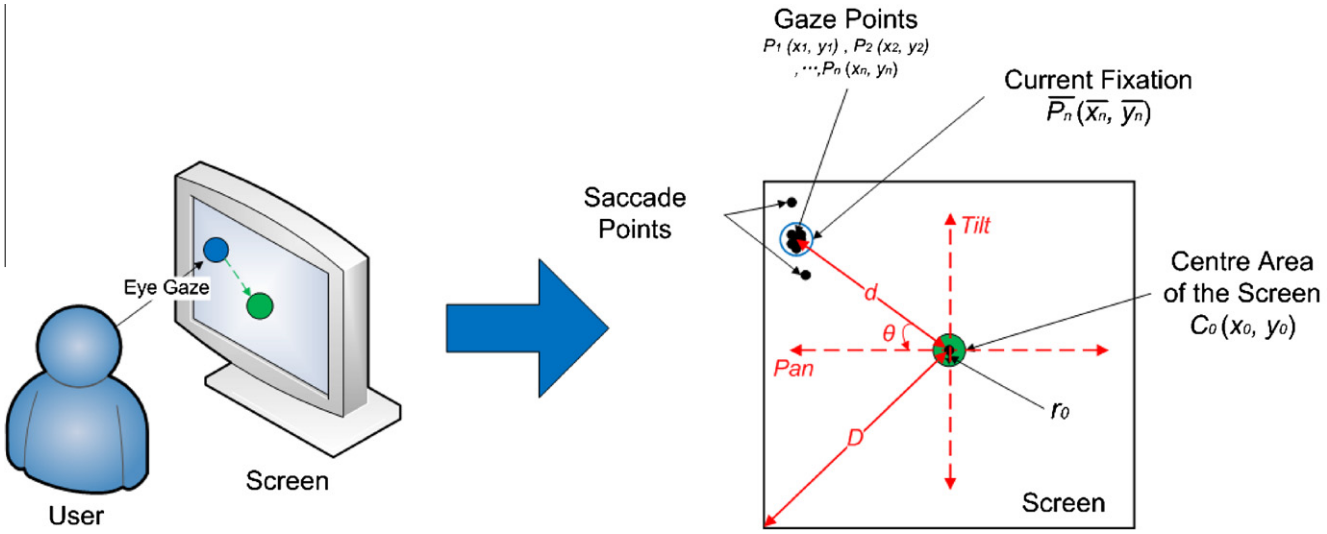

Fig. 4. Design of gaze-driven remote camera control.

where $CAM_{velocity\_max}$ represents the maximum velocity of the camera. In our default implementation, $CAM_{velocity\_max} = 30°/s$; FG is a liner function gain calculated as a ratio between the distance of the current fixation to the centre d and the maximal distance on the screen to the centre D. It is used to translate the ratio to the corresponding camera velocity proportionally, which means the further you look at from the centre, the faster the camera moves towards the centre.

4. The corresponding camera velocity on both pan $CAM_{velocity\_current\_pan}$ and tilt $CAM_{velocity\_current\_tilt}$ directions are calculated as follow:

$$\begin{cases} CAM_{velocity\_current\_pan} = CAM_{velocity\_current} \cdot \cos\theta \\ CAM_{velocity\_current\_tilt} = CAM_{velocity\_current} \cdot \sin\theta \end{cases} \quad (6)$$

The camera motion will keep following the user's current fixation direction, if its position is not in the centre area of the screen. The overview of the entire process is that wherever the user focuses their visual attention in the video stream, the camera will always bring that to the centre of the screen. Therefore, the user will not feel that they are actually performing much "deliberate control" of the camera movements.

5. Prototype implementation

The prototype system contains two major parts: the user end and the remote camera site, in between can be a standard network connection. The overall system structure is illustrated in Fig. 5.

At the user end, we integrated the FaceLAB[3] (V4.5) eye-tracking system (laptop version) into our prototype, which provides the real-time gaze tracking at a 60 Hz frequency without the use of markers. This avoids the need to make the user wear any specialized devices, offering comfort and flexibility. Head mounted trackers can provide more accuracy and a higher tracking frequency but they are not comfortable to wear for long.

We used a Dell Precision Work Station with standard Window XP operating system installed as the main PC. The FaceLAB eye tracker was connected to the main PC through a local network for transferring the real-time raw gaze data. The FaceLAB Client Tools SDK was installed on the main PC, called by the gaze-driven camera control code for receiving the raw data from the local network. The control code translates the raw gaze data into corresponding camera control commands as we explained in the previous section, and sends the real-time commands to the remote camera through the external network. The laptop-based eye tracker shared the user screen for eye tracking on the main PC, as the user would only be watching the video stream from the remote camera on the user screen. The gaze-driven camera control code and other relevant software integrations were all implemented in Visual C++.

On the remote site, we used the Pelco ES30C[4] (the same mode of camera has been used in the real rock breaking setting) as the remote camera to be controlled in the prototype system with the capability to perform pan and tilt functions simultaneously. It was connected to the user end through an external network, transferring the live video stream back to the user and also receiving the control commands to carry out the relevant camera movements. The camera is able to provide 360° continuous pan rotation and a tilt range of +33° to −83° from horizontal.

Both gaze data processing and camera operation happen simultaneously as everything is being operated in real-time, i.e. the system does not operate the camera control on an iterative detect-gaze and move-camera process which would result in jerky movements for the camera. The camera motion is not able to match the original gaze tracking frequency. Therefore, fixations are used as smoothed inputs to synchronize the frequencies of sending and receiving control commands for the camera. Apart from the introduced 200 ms time interval for detecting each fixation, adding camera operation latency and network data transferring delay, the entire latency for the system is less than 250 ms, which has proved to be fairly tolerable in the user evaluation.

6. Evaluation

We wished to investigate how well the gaze-driven camera control could perform in a model of a real-world hands-busy setting, in comparison to the conventional control. Therefore, we implemented a joystick based camera control as an example of a conventional control interface in teleoperation by using a standard Logitech wireless gamepad.

For the joystick control, the user can control the remote camera motion by pushing the left joystick on the gamepad. The camera

[3] http://www.seeingmachines.com/product/facelab/.

[4] http://www.pelco.com/products/.

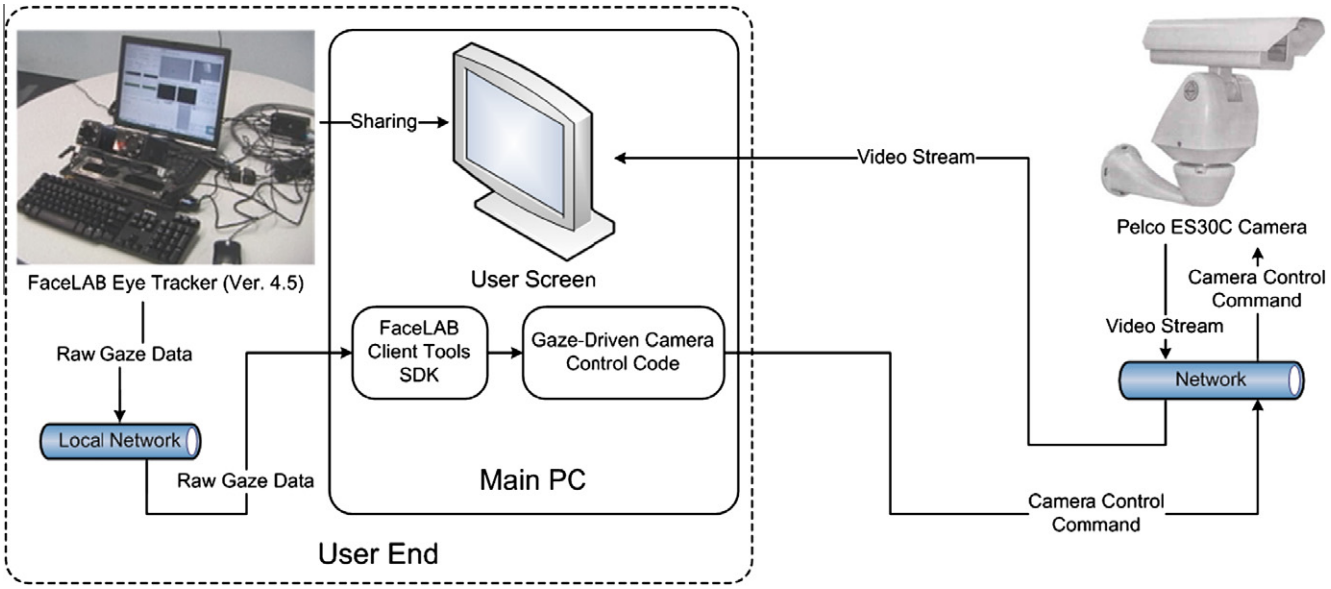

Fig. 5. System diagram for the implementation of the gaze-driven camera control prototype.

moves along the same angle as the user moves the joystick, and it will stop its movement if the user releases the joystick.

6.1. A modeled hands-busy task using functional physical modeling

We had very limited access to the real rock breaker equipment, we used the concept of *functional physical modeling* for our experimental task design, by which we could model many of the properties of the example setting by using another physical model. Our goal is improving teleoperation camera control, which is motivated by an example real world setting of the rock breaker. We designed our experimental setting to be similar in its key properties to improve the likelihood that our results can later be implemented in such real world settings.

A *functional physical model* (Gedeon and Zhu, 2010) is defined as a set of equipment that has been designed to reproduce some specific properties of another setting in a real-world context for the purpose of evaluation. Generally, such a model would be appropriate for cases where any specific property is difficult to measure within the original setting, or there exists difficulties in access to either the original equipment or the real operators or even both, or some other unavoidable impediments to reproduce experiments with the original setting.

After extracting the rock breaking properties which may affect performance and productivity in that task, we decided to model this hands-busy setting by using a physical game analogue: playing a re-designed foosball game with two handles. We recruited university students as experimental subjects. We chose to construct our functional physical model primarily on such a game based task was to include the competitiveness and engagement we observed among the real operators performing operations in the example real world setting, with daily and weekly tallies of points and so on. Since we had also observed that university students became engaged and competitive in games, we believe that our design could be an appropriate model with the advantage of being more compelling and interesting to our student based subjects than simulating an abstract, boring and industrial-like control task.

Note that we also avoid any optimizations to the functional physical model which could not be done in the example setting, so for example we do not use preset camera positions since they would not be useful in the example real world setting.

We considered using a 3D simulation of the rock breaker. It is possible that students (as the operators were not available) would have developed competitive behavior. To retain realistic properties of the example real world setting such as camera motion lag, the 30 s wait while the sprinklers settle the dust and so on, would have made it quite a frustrating game. Thus some justification was required so we opted for the same model camera in a functional physical model.

Users have direct control over the game through physical handles in our model, but in the example real world setting they control the robot arm via teleoperation. The difference between the re-designed model and the example real world setting is that the model has no lag on the hand control while the rock breaker joysticks have some lag because of teleoperation over distances. However, we do not believe this is important as the camera is not moving when the handles or the jackhammer is used.

The re-designed foosball model is shown in Fig. 6, which involved a number of changes and re-constructions from the original small tabletop size two-player foosball table:

1. In order to make it a single-player game, one pair of handles on one side have been removed from the table, and the right side goal was blocked by a sloping surface to return the ball.
2. A pair of plastic wheels with metal weights mounted were attached on the end of the other two handles to make them self-centered generally analogous to the joystick control mechanism. Both the wheels and joysticks will self centre when the hands are removed.
3. The table surface has been modified to be contoured with specific channels by constructing various shapes of ridges so that if the user kicks the ball "too hard", it will tend to roll back; while the user kicks the ball "just right", it will roll towards the goal. The table also slightly slopes, away from the goal, so that kicks which are "too soft" will roll back to a rest position.
4. The sloping surface under the middle man on the left is quite slight. Thus it is rare for the ball to stop there. The left and right stopping spots for the ball are positioned differently, the left ones being closer together.

As to the actual experimental setting (see Fig. 7), we used a standard 19″ monitor as the major user screen with a resolution

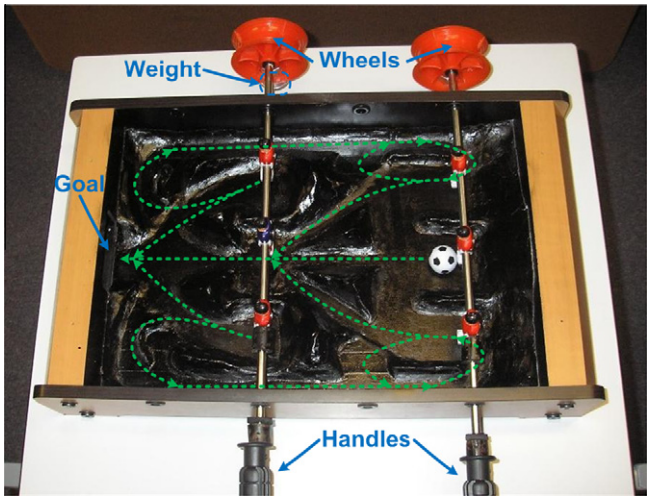

Fig. 6. Re-designed foosball table as a functional physical model.

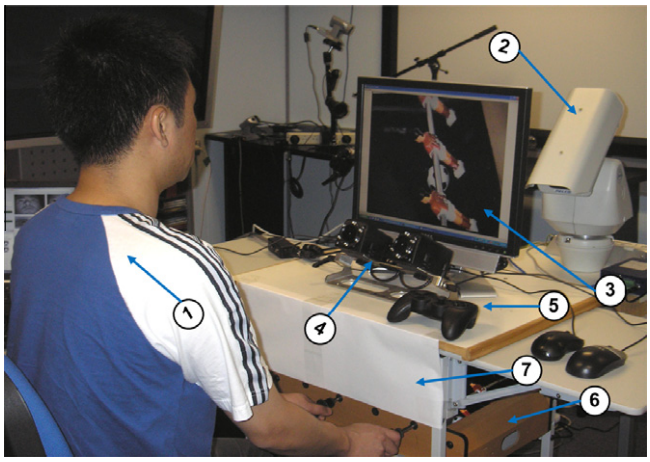

Fig. 7. Experimental setting: (1) participant, (2) remote camera, (3) screen view of video stream, (4) eye tracker, (5) gamepad as joystick control, (6) re-designed foosball game, (7) covers to obscure participant's direct view of the foosball table.

of 1280 × 1024 pixels, showing the video stream from the remote camera to the participant. The re-designed foosball table was placed under the monitor with several covers attached on the near side to obscure the participant's direct view of the foosball table so that the view of the setting was via the camera and screen only.

As we mentioned in the introduction, when the operator is performing the rock breaking operation, they actually only focus on a very small region of the whole bin, which is mostly the area of a rock on which the operator is trying to specify the spot for breaking by moving the robot arm and firing the jackhammer on the tip. To optimize this exploration process, the operator always makes the camera zoom close in so they are able to acquire enough details of the target rather than making the camera zoom in and out all the time. They then just use the pan and tilt controls on the remote camera to find the spot for breaking.

In order to match this condition, we set the camera zoom to a level to only have a partial view of the field in our experiment (see (3) in Fig. 7), leaving the camera pan and tilt control to the participant. It effectively makes participants keep performing the camera control to find the ball throughout the whole experimental period, whenever the ball is out of the current area of vision. This design produced control behavior which appeared similar to the way we observed operators conduct their rock breaking task, thus, any benefits we find for our camera control prototypes are not likely to be less in the example real world setting.

6.2. Participants

A total of 24 undergraduate students (mostly first-year undergraduates) voluntarily participated in the user evaluation and successfully performed the experiment, including 19 male and 5 female. The age of participants ranged from 18 to 23, with a mean of 19.6 years old and *SD* = 1.5.

All participants were regular computer users (at least 2 h per day) with video game experience of using a joystick-based interface, but none of them had any prior experience using an eye tracker or our re-designed foosball game. Several participants wore glasses, the rest had normal vision without any correction, and their eye gaze could be calibrated all successfully.

6.3. Experimental design

The experiment was conducted by using a repeated measures within-subject design so that all the subjects participated in all conditions of the experiment. The major independent variable was the *camera control method* by which we compared gaze-driven control with joystick control.

The order effect was eliminated by switching the order of the camera control methods. Both objective measures and subjective measures were used in the user study.

6.4. Procedure

Participants took part in the evaluation individually. Prior to starting the experiment, participants were given a short oral presentation (around 5 min) about the user study. The context included an introduction to the system, instructions on how to control the remote camera by using the gaze-driven control and the joystick control respectively, and how to play the re-designed foosball game. The objective of their play was to score as many goals as they could during each camera control trial. All the participants were required to confirm an understanding of these introductions and the requirements of the experimental task.

After the completion of the oral introduction session, participants started the experiment directly, no pre-training period was provided before the formal experiment. For each control method, participants had 5 min to play the re-designed foosball game. An extra 3–5 min were spent on the calibration of each participant's eye gaze before they started the gaze-driven control trial.

The video stream from the remote camera for each participant using different camera controls were recorded respectively. In addition, their entire experimental period was also recorded by another video camera for further observations. For the objective measures, the number of goals and kicks each participant achieved was recorded through checking against the video records. A kick is a purposive movement of a foosball man as controlled by the handles when it properly engages the ball by moving it some detectable amount. By purposive we mean that if a man engages and moves the ball while it is not visible on screen then it is a random or accidental movement and not purposive. As we record the timing of kicks and record the screen view, purposive kicks are simple to determine.

Once the foosball game under both of the camera control methods had been finished, we collected the participant's qualitative feedback on the prototype by using a questionnaire with a 5-point Likert scale, rating from 1 (strongly disagree) to 5 (strongly agree) and a short interview, in which they compared their experiences with different control methods across several criteria as subjective

measures, including naturalness, required consciousness, distraction and time to get used to each control method.

7. Results

We report the results of the user evaluation through objective measures and subjective measures respectively. By conducting statistical analysis on both of the measures, we demonstrate the comparisons of user performance and preference between the gaze-driven control and the joystick control quantitatively and qualitatively.

7.1. Objective measure results: user performance on goals and kicks

The major objective measures are according to the analysis of the number of goals and the number of kicks each participant achieved in the corresponding camera control trail.

A Paired *T*-test showed a highly significant difference in scored goals between gaze-driven control and joystick control, $T(23) = 4.27$, $p = 0.000143$. Fig. 8 shows the overall mean goals for each camera control method, and we can clearly see that using gaze-driven control ($M_{goals\_gaze} = 5.83$, $SD_{goals\_gaze} = 1.83$) in the experiment on average participants significantly scored more goals than using traditional joystick control ($M_{goals\_joystick} = 3.71$, $SD_{goals\_joystick} = 1.88$).

In addition, the Paired *T*-test for mean kicks showed very similar result to the previous mean goals analysis. A highly significant difference on the number of mean made kicks between using gaze-driven control and joystick control was also found, $T(23) = 4.33$, $p = 0.000125$. Furthermore, from the mean comparison illustrated in Fig. 9, it shows that the number of kicks participants made using gaze-driven control ($M_{kicks\_gaze} = 27.08$, $SD_{kicks\_gaze} = 4.05$) on average is significantly more than the number of kicks made by using joystick control ($M_{kicks\_joystick} = 23.46$, $SD_{kicks\_joystick} = 4.49$).

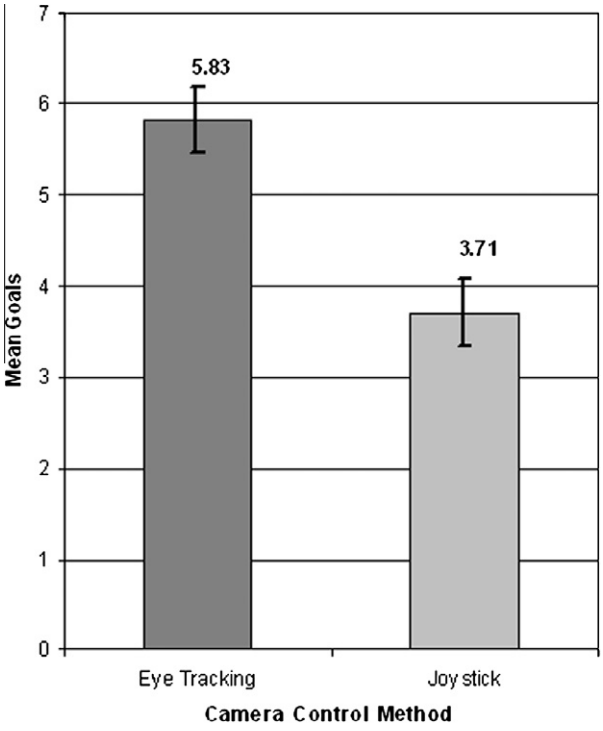

Fig. 8. Mean goals for each camera control method.

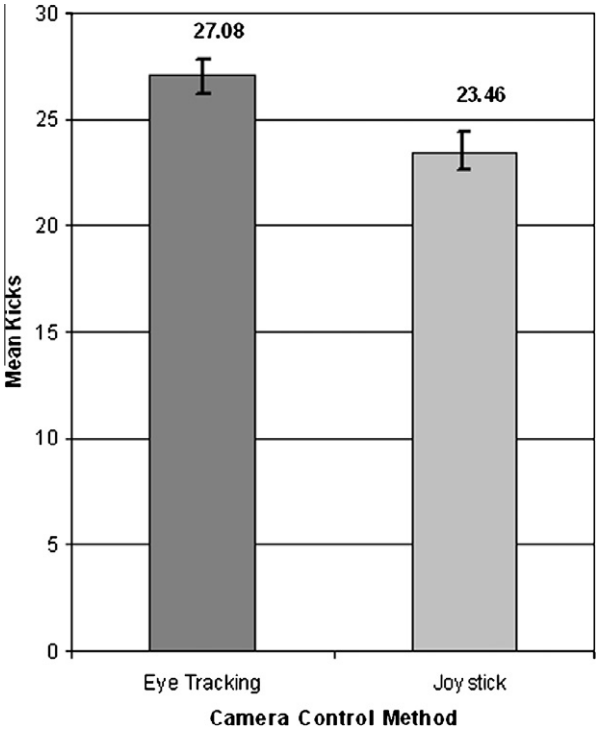

Fig. 9. Mean kicks for each camera control method.

7.2. Subjective measure results: user preference and feedbacks from questionnaire and interview

The questions and results of all participants from our questionnaire regarding "naturalness", "time to get used to the control", "consciousness" and "distraction" for each camera control method are depicted in Fig. 10.

Almost all the questions show significant results favoring the gaze-driven control, including Q1, Q3 and Q4. Although there is no significant difference in the results of Q2, in fact the mean result of gaze-driven control ($M_{Q2\_gaze} = 4.21$, $SD_{Q2\_gaze} = 0.72$) is still slightly better than the joystick control ($M_{Q2\_joystick} = 3.83$, $SD_{Q2\_joystick} = 1.05$) regarding the user feedback on time to get used to the corresponding control.

At the end of the questionnaire, participants were asked to state an overall preference of these two camera control methods according to the experience in the experiment. A majority of the participants, 18 out of 24 (75.0%) preferred to use the gaze-driven control, and the rest 6 (25.0%) showed their preference of using the joystick control.

From the short interview conducted at the end of the study and the comments participants made when filling in the questionnaire, most of them felt that using gaze-driven control was quite effective for resolving the hands-busy problem in the experiment. Compared to the conventional joystick control, it was more convenient and flexible, required less physical movements and attention.

8. Discussion

We further discuss the results shown above in a more detailed way. Instead of introducing more formal research findings of our user study, we would like to present observations across all the relevant information obtained at this point.

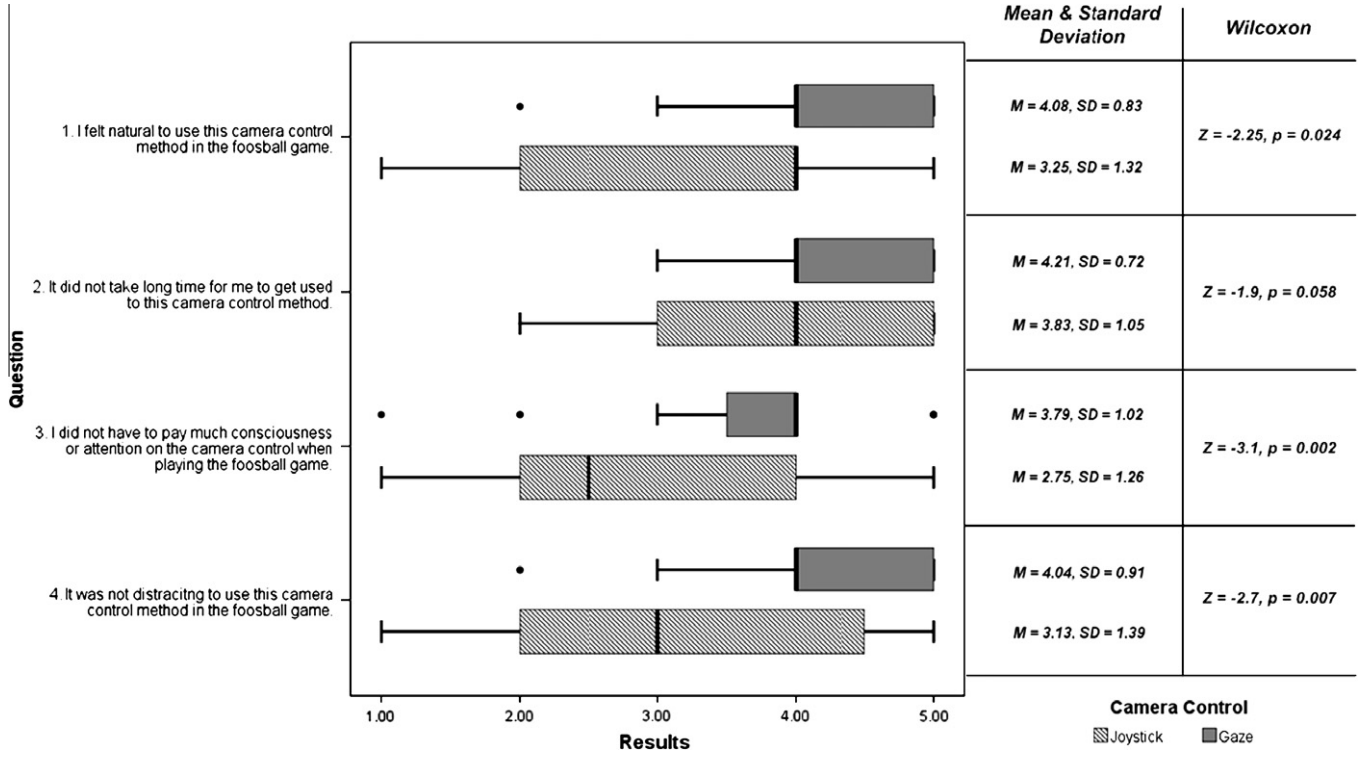

Fig. 10. Results from questionnaire (5-point Likert scale: 1 – strongly disagree, 2 – disagree, 3 – neither agree not disagree, 4 – agree, 5 – strongly agree) feedback for gaze-driven and joystick camera control, showing boxplot, values of mean and standard deviation and results of Wilcoxon test for each question.

8.1. Further discussions and observations on results

From the results of the objective measures, it has clearly indicated that the gaze-driven control performed quantitatively better than the joystick control for most of the participants even without a pre-training period in our modeled hands-busy experiment. More valid kicks and more goals were produced, which followed the general anticipation of the game-like task that more kicks would lead to more goals. This in fact tells us that using the gaze-driven camera control, participants were able to obtain significantly more opportunities to kick the ball and score more goals, since they were not distracted by using another control interface.

Moreover, the user assessment was consistent with the objective results. The gaze-driven control outperformed the joystick control through all the criteria we selected. Particularly for Question 1 regarding the naturalness of the control, most of the participants clearly felt that they could naturally pick up the gaze-driven control without actually practicing the control mapping beforehand. This has been reflected on the statistical significance from the Question 1 user data analysis.

We did not find any significant difference on the user data of Question 2 regarding the time to get used to the control. The reason could be all the participants already had previous experience of using a gamepad for video games, so it might not take a long time for them to get familiar with the control mapping of using the joystick control as it was just a very standard mapping for controlling or navigating like most control configurations used in video games or virtual environments they had experienced before.

In addition, from the results especially for consciousness of control (Question 3) as well as the relevant comments from the interview, we can conclude that there was not much consciousness or attention required for participants to control the camera motion by using their gaze, when simultaneously playing the foosball game. Also, most of the participants directly mentioned that using this gaze-driven control allowed them to pay significantly more attention on the task they were doing with their hands, and therefore it was actually not necessary for them to "think" much about how to adjust the camera view to obtain enough visual information for kicking the ball. This effect clearly reflects back to the original notion of designing such a gaze-driven camera control, which was considered to have the potential promise of not offering much obvious experience of "deliberate control".

As we mentioned before, the entire gaze-driven control just followed a very simple design principle: "*Whatever you look at the video, the camera will bring it to the centre of the screen.*" Hence, there was no specific control configuration to be adapted to by the participants. In contrast, the camera view was automatically adjusted based on the participant's current visual attention. Thus, it did not require conscious attention allowing full attention to be focused on the primary task. Such effect has also been reflected in the statistical results of distraction in Question 4.

8.2. Existing issues and arguments

According to the experimental observations and the comments from the participants, the major issue with the gaze-driven camera control was the unreliability of the gaze tracking process. Several participants commented that it was still a bit sensitive and unreliable, as they were actually not able to have a completely free interaction. They noticed that when they occasionally moved their head direction along with the gaze unconsciously (it actually happened often for most of the participants), the gaze tracking quality was reduced or sometimes the tracking would be lost if they moved their head a bit further away, which would directly affect the control quality of the remote camera. This was the major reason a subgroup of the experimental population (6/24 = 25%) ended up with the overall preference to the joystick control as reported by these

subjects. Improvements in eye tracking hardware and software will reduce the effect of unreliability of the tracking process.

During the experiment as well as the post-experimental video record checking, we observed that a few participants attempted to score more goals by optimizing their control behavior. Participants were using one hand on the joystick to control the camera, using the other hand swapping between the two foosball handles. This might be a difference in our functional physical model. That is, in the example real world setting, to carefully specify a breaking spot on a target rock required more precise positioning than required by our foosball model. In the rock breaking setting, when the proper camera view was achieved then the operator could carry out the final tip positioning. These are limitations of our work.

However, we believe and argue the mapping of our functional physical model to the rock breaking setting is quite plausible, as the re-designed foosball model matches many properties of the example setting. These include the similar control mechanism of the devices, similar operation process to complete the task, similar objective of the operation, competitive working condition, in a hands-busy situation. Moreover, the foosball game is engaging for students and participants did report that they enjoyed the experiment. Thus there is some likelihood that our gaze-driven control could have similar benefits in the example real world setting as demonstrated in the evaluation.

9. Conclusions and future work

Motivated by a common hands-busy problem existing in current teleoperation settings and considering a real world complex mining industrial task as the example to design our research scenario, we present a novel gaze-driven remote camera control as an effective solution with a developed prototype system.

In order to cope with severely limited access to the example real world setting and operators for conducting user evaluation, we used a *functional physical model* to reproduce some key properties of the example real world setting by the use of a re-designed foosball game. A user study with 24 university participants on our gaze-driven camera control in comparison to a conventional joystick control using this functional physical model has been conducted through both objective measures and subjective measures.

From the objective results, we show that using the gaze-driven control, participants performed significantly better than using the joystick control without pre-training in the modeled hands-busy experiment. In addition, the subjective results also reveal clear evidence that the gaze-driven control significantly outperformed the joystick control through almost all the criteria we selected.

A few existing issues of the gaze-driven control and the way to use the functional physical model (the foosball game) for the experimental task design have also been discussed, arguing this model is plausible to test in the example real world setting, and therefore the results obtained from our user study may be applicable and beneficial in that setting.

The future directions can be exploring more natural human interaction based design prototypes, or multi-model designs by combining gaze and other types of interaction techniques particularly for user control problems in teleoperation, and the investigation of improved functional physical models for user studies.

Acknowledgements

The authors would like to express their appreciation to all the students that voluntarily participated in the user evaluation. The authors also thank Chris Gunn, Matt Adcock and Leila Alem from the CSIRO ICT Centre, and Jock Cunningham and Eleonora Widzyk-Capehart from the CSIRO Earth Science and Resource Engineering for their great help and valuable suggestions.

This research was supported by the Transforming the Future Mine theme under the CSIRO National Mineral Down Under (MDU) Research Flagship.

Appendix A. Supplementary material

Supplementary data: a video component (comparing joystick-based remote camera control with gaze-driven control for a modeled hands-busy task) associated with this article can be found, in the online version, at doi:10.1016/j.intcom.2010.10.003.

References

Cohen, C.J., Conway, L., Kiditschek, D., 1996. Dynamical system representation, generation, and recognition of basic oscillatory motion gestures. In: Proceedings of 2nd International Conference on Face Gesture Recognition, pp. 60–65.

Duff, E., Caris, C., Bonchis, A., Taylor, K. Gunn, C., Adcock, M., 2009. The development of a telerobotic rock breaker. In: Proceedings of 7th International Conference on Field and Service Robots (FSR 2009), pp. 1–10.

Fong, T., Thorpe, C., 2001. Vehicle teleoperation interfaces. Autonomous Robots 11 (1), 9–18.

Gedeon, T.D., Zhu, D., Mendis, B.S.U., 2008. Eye gaze assistance for a game-like interactive task. International Journal of Computer Games Technology 2008 (623725), 1–10.

Gedeon, T.D., Zhu, D., 2010. Developing a natural interface for a complex task using a physical model. In: Proceedings of 2nd IEEE International Conference on Intelligent Human–Computer Interaction (IHCI 2010).

Goh, A.H.W., Yong, Y.S., Chan, C.H., Then, S.J., Chu, P.L., Chau, S.W., Hon, H.W., 2008. Interactive PTZ camera control system using WII remote and infrared sensor bar. In: Proceedings of World Academy of Science, Engineering and Technology, pp. 127–132.

Hainsworth, D.W., 2001. Teleoperation user interfaces for mining robotics. Autonomous Robots 11 (1), 19–28.

Hughes, S., Lewis, M., 2004. Robotic camera control for remote exploration. In: Proceedings of 22nd ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2004). ACM Press, pp. 511–517.

Isokoski, P., Hyrskykari, A., Kotkaluoto, S., Martin, B., 2007. Gamepad and eye tracker input in first person shooter games: data for the first 50 minutes. In: Proceedings of 3rd Conference on Communication by Gaze Interaction (COGAIN 2007), pp. 11–15.

Istance, H., Bates, R., Hyrskykari, A., Vickers, S., 2008. Snap clutch, a moded approach to solving the midas touch problem. In Proceedings of 2008 Symposium on Eye Tracking Research and Applications (ETRA 2008), pp. 221–228.

Jacob, R.J.K., 1991. The use of eye movements in human–computer interaction techniques: what you look at is what you get. ACM Transactions on Information Systems 9 (3), 152–169.

Kobayashi, H., Kohshima, S., 2001. Unique morphology of the human eye and its adaptive meaning: comparative studies on external morphology of the primate eye. Journal of Human Evolution 40 (5), 419–435.

Kumar, M., Paepcke, A., Winograd, T., 2007. Eyepoint: practical pointing and selection using gaze and keyboard. In: Proceedings of 25th ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2007), pp. 421–430.

Saito, S., 1992. Does fatigue exist in quantitative measurement of eye movements? Ergonomics 35 (5/6), 607–615.

Salvucci, D.D., Goldberg, J.H., 2000. Identifying fixations and saccades in eyetracking protocols. In: Proceedings of 2000 Symposium on Eye Tracking Research and Applications (ETRA 2000), pp. 71–78.

Smith, J.D. Graham, T.C.N., 2006. Use of eye movements for video game control. In: Proceedings of 2006 ACM SIGCHI Conference on Advances in Computer Entertainment Technology (ACE 2006), No. 20.

Tall, M., Alapetite, A., Agustin, J.S., Skovsgaard, H.H., Hansen, J.P., Hansen, D.W., Mollenbach, E., 2009. Gaze-controlled driving. In: Proceedings of 27th ACM SIGCHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI 2009), pp. 4387–4392.

Tanriverdi, V., Jacob, R.J.K., 2000. Interacting with eye movements in virtual environments. In: Proceedings of 18th ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2000), pp. 265–272.

Wang, S., Xiong, X., Xu, Y., Wang, C., Zhang, W., Dai, X., Zhang, D., 2006. Face tracking as an augmented input in video games: enhancing presence, role-playing and control. In: Proceedings of 24th ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2006), pp. 1097–1106.

Yarbus, A.L., 1967. Eye Movements and Vision. Plenum Press, New York.

Yamaguchi, K., Komuro, T., Ishikawa, M., 2009. PTZ control with head tracking for video chat. In: Proceedings of 27th ACM SIGCHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI 2009), pp. 3919–3924.

Yanco, H., 1998. Wheelesley: a robotic wheelchair system: indoor navigation and user interface wheelchairs. Assistive Technology and Artificial Intelligence, 256–268.

Zhai, S., Morimoto, C., Ihde, S., 1999. Manual and gaze input cascaded (magic) pointing. In: Proceedings of 17th ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 1999), pp. 246–253.

Zhu, D., Gedeon, T.D., Taylor, K., 2009. Keyboard before head tracking depresses user success in remote camera control. Proceedings of 12th IFIP TC13 Conference on Human–Computer Interaction (INTERACT 2009), vol. 5727. LNCS, pp. 319–331.

Zhu, D., Gedeon, T.D., Taylor, K., 2010. Natural interaction enhanced remote camera control for teleoperation. In: Proceedings of 28th ACM SIGCHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI 2010), pp. 3229–3234.