INTERNET CONTROLLED ROVER

By Farzad Eskafi, Bryon Ross, Ling Xiao
{feskafi ,bryonr, xiaoling} @ uclink.berkeley.edu

ABSTRACT

In this paper, we describe the implementation and design of an Internet Controlled Rover, a mobile device that transmits a high quality video feed, as well as sound, over high bandwidth internet. The system consists of a server, a micro-controller board, a remote controlled rover, a wireless web cam and a wireless receiver. We describe system specifications and their functions in detail and the difficulties experienced while creating the rover. Finally, we describe possible future work and system improvements.

1. INTRODUCTION/BACKGROUND

Traveling is expensive. In today's world, where time is money, people do not have the luxury of long-term trips. Even short vacations require a considerable amount of money, time, and planning. People spend long hours in front of computers or in labs, when instead they should be roaming the world. We have a solution to this problem.

Our idea is to have small remote controlled devices roaming around interesting tourist, shopping, and adventure sites around the world. An online user who 'rents' the unit for a certain amount of time will control each remote controlled device, and the device will be able to transmit high quality video and sound. Thus users could view environments that they would otherwise be unable to enjoy from the comfort of their homes. The rovers will each be constrained by operational criterion specified by each operating environment, such as predefined paths, tracks, depth, collision, speed, acceleration and range. Yet the users still experience the environment around it as if they were there.

The device has several parts. The main parts are Video, Rover Control, Sound and Chat. Each main part has its own software and hardware part. The basic design and architecture is explained in *Architecture Overview*. The implementation is explained fully in *Implementation Details*. We thoroughly tested our system. The testing procedure is explained in *Evaluation and Testing*. And finally *Future Work* tells the reader what other possible improvements could be done on the rover.

2. ARCHITECTURE OVERVIEW

The basic design and architecture of our project is comprised of four primary components: control, video, sound, and chat.

2.1 Control System

The control system deals with the transmission of control signals from the user to the rover. These control signals determine a rover's actions, such as "move forward" or "move backwards and to the right". Only one user may control a rover at a time. In our implementation, control is given on a first-come, first-serve basis.

2.2 Video System

The video system retrieves images from the rover's on-board camera and transmits them to a user. These images are displayed to the user as a video stream. Multiple users may access an active rover's video stream. The video system includes a simple image processor, which can be used to convey environmental information, and the ability to save still images onto a local storage device.

2.3 Sound System

The sound system receives sound from a microphone on a rover and transmits it to a user who is viewing that rover's video stream. While sound often enhances the user's virtual environment, it can also become burdensome to both the user's experience and the underlying system. For this reason, the user may optionally disable sound.

2.4 Chat System

The chat system allows users to communicate with one another while controlling or viewing a rover. This is the only primary component that is independent of the hardware, serving only to allow interaction between users, to enhance the user experience. Reticent users may optionally choose to opt out of chatting without compromising the video or control systems.

3. IMPLEMENTATION DETAILS

Except chat, every system is composed of four major components: rover hardware, server hardware, client software, and server software. Chat has no hardware components.

3.1 Control System

Rover hardware: Radio Shack "Monster Patrol" Radio Control Truck

Server hardware: Infineon SK-167CR 16-bit Microcontroller

On the server side, control signals (commands) received from the user are relayed to a microcontroller through a serial port. The microcontroller converts these commands into electrical signals, which are fed through wires into the remote control included with the rover. The net effect is the transformation of a standard RC car into a computer-controlled car.

On the client side, a user wishing to gain control of a certain rover sends a request to the server. Upon request, the server will attempt to establish a connection to the microcontroller. If another user is controlling the rover, the server will fail to activate the microcontroller's serial port and the new user will fail to gain control. If the user does gain control, he or she may use a graphical user interface of directional arrows or customizable keyboard interface to send control signals for a certain rover to the server. The user will maintain control until he or she terminates the control system or the client/server connection fails.

Commands are currently limited to the numbers 1 through 9, each representing the corresponding numeric keypad direction (1 is left-reverse, 8 is forward, etc.).

3.2 Video System

Rover hardware: D-Link DCS-1000W 2.4GHz Wireless Internet Camera

Server hardware: Linksys Wireless USB Network Adapter

When a user attempts to access a rover's video, the server will activate that rover's camera (if not already active) and start a thread that will retrieve images from the camera as long as at least one person is accessing the video. The server will then forward these images to the user, to whom they are displayed as a video-like stream of JPEG images.

The image refresh rate is primarily determined by the processing speed of the client computer and network speed. The image retrieval thread will retrieve a new image from the camera whenever a client that has already retrieved the previous image requests another. The client will only request another image after it has completely rendered and displayed an image.

A client can display multiple instances of a several rovers simultaneously. Also, many clients can view the same rover's video stream simultaneously. Since all images are intermediately stored on the server by the image retrieval thread, the rover-to-server network is not overly taxed by image retrieval, but still provides the client with the most current images.

3.3 Sound System

Hardware: 2 Radio Shack Two-way Personal Radios (1 on server, 1 on rover)

There are three stages associated with the sound system: the rover side, the server, and the client.

On the rover side, we used a personal radio that has been permanently set on the transmit mode. In order for the user to hear a sound, the sound is fist picked up by the microphone of the rover radio and transmitted to its partner, which is connected to the microphone input of the server computer. Then the sound server thread on the server computer will convert the microphone input into a sound stream that is sampled at 44100 times per second, with each sample being 16 bits and 2 channels. The sound server thread will chop the sound stream input into 4096 sized blocks, which are then transmitted using TCP/IP through the Internet to a corresponding sound client thread that is running on the client computer. The client sound thread will take the 4096 bytes of payload of the TCP/IP packet and translate it back into the original sound stream, with the same parameters as the sound server. Then finally the sound stream is fed into the output stream of the client computer, such as speakers.

3.4 Chat System

There are three stages associated with the chat system: the sender user, the server, and the receiving users.

When a user initially logs on to the chat server of a particular rover, the main chat server thread associated with that rover will instantiate a new communications thread for that client. The communications thread will be responsible for transmitting messages to and receiving messages from the client computer. There is one main chat server thread associated with each rover, and each main chat server thread listens in on a distinct server port to distinguish different rover chat rooms. To log on to the chat server, the client chat program will first send the user's username, and wait until the server sends a response notifying the client that the server is ready to receive and transmit.

There is a chat client thread running on the user computer that will read in user input and send it to the communication thread associated with that user on the server computer. The communications thread has a member class that keeps track of all current communications threads running for the rover. Once the communications thread receives a message from the user, it will first tag the user's username to the beginning of the message and then use above-mentioned class to write the message to all the communication threads. Each communication thread will subsequently send the message via TCP/IP to the client computer's chat thread, which will then display the message on the chat window of the client computer.

3.5 Bar Detector

The bar detector scans through images produced by the video camera counting the number of bars found in the image. It is a simple attempt at video bar code recognition, thus allowing the environment to convey additional information to the client.

When passed an image to process, the bar detector will interpret the image as black and white, by summing up the RGB values for each pixel, and comparing it to a certain standard (200 for the demo). If the sum of the RGB values were above the standard, the color would be considered black, and if the color were below the standard it would be considered white.

To find the bars in the image, the bar detector will traverse the image from left to right, one pixel at a time, looking for bars that change from black to white, or white to black. A 'bar' in this purpose is a strip that has the pixels +X (e.g. 5) pixels above and –X (e.g. 5) pixels below which are the same color as the pixel in the center. When a bar transition is found, the bar detector will count the width of the bar by counting the number of pixels traversed since the transition and before either another transition or degeneration into a non-bar formation. The resulting bar widths are stored in an array, starting at element zero.

Once the image has been traversed, the bar detector will iterate through the array looking for the maximum number of consecutively similar width transitions – transitions that have width within a few pixels of each other are considered similar width. Since in a natural environment the number of similar width transitions will be small, if the number of transitions is above a certain limit, we can guess that there have been artificially introduced bars. Since there are 2 transitions for every bar, the number of actual bars that are represented by the number of transitions found will be the final step in bar counting.

Since the above process takes a fairly long time, the bar detector will only traverse the middle of the image, and is run in a separate thread that will grab a new image from the image client whenever it has finished processing. Also because of the estimating nature of the algorithm, multiple runs of the bar detector is required to make a good guess at the number of bars actually present.

4. EVALUATION AND TESTING

We completed two major types of testing on the rover prior to public display: functionality and quality of service. Functionality testing evaluated the rover's ability to work properly. This involved ensuring that all available user options were implemented correctly and testing the maximum functional distance between the rover and the server. We found two major functionality failures while testing the rover. First, the control signals are often unable to transmit through large amounts of solid matter. If several solid walls were placed between the rover and the server, the rover would stop responding to user input. This is due to the 5V maximum output of the control system's microcontroller, which drives the transmitter. (The transmitter is designed to work optimally at 9V.) Due to monetary and time constraints, we were unable to fix this problem. The second problem is power loss to the camera after a high-speed collision. The traumatic force causes the camera to reset, thus interrupting its connection to the server. Often the server and camera can automatically recover from this error, but in some cases, the server administrator must manually reinitialize the rover's camera.

Quality of service testing assured that users enjoyed their experience with the rover. We allowed several users to connect to our rover from existing broadband Internet connections (such as DSL), as well as over the local area network (using gigabit Ethernet). We found the image refresh rate over the existing Internet connections was sufficiently slow (about 2 frames/second) to interfere with user enjoyment. Also, sound was broken and filled with static. Over the local area network, the refresh rate was significantly higher (about 15-30 frames/second), sound was clearer, and users tended to enjoy their experience. A few users were initially disoriented by the control system, but only one was unable to successfully maneuver the rover after a few minutes.

Despite a few minor concerns, we feel the rover successfully met the requirements set forth at the start of the project.

5. FUTURE WORK:

Our rover could always use improvements. In this section, we discuss some of the possible improvement that could be done with our system. As will be noted, our code is able to handle some of these improvements.

5.1 Multiple or Ratable Cameras

The Remote Control Device contains one camera. As explained above, this camera transmits high quality video. As a result, the Internet user could only get one view of the environment. He or she could only see what's in front of him/her. To let the user get a much better view of the environment around it, we propose that we equip the rover with multiple cameras, mounted on multiple sides. Another possible solution is to mount several tiny cameras all around the rover, so that each camera transmits high video to a server. Upon receiving the images, the server puts them together into one high quality picture and sends it to the client, where the user is awaiting an image. Alternatively, the user could be allowed to manipulate the camera angle. This provides a more cost-effective and versatile viewing apparatus.

5.2 More interaction with the Environment

The rover can only move on flat ground with no major obstacles on it. If a situation occurs when there is an obstacle in front of the rover, it simply cannot remove the obstacle and continue the tour. To let the rover interact more with the environment, we propose that we should add appendages to the rover, so it could go up and down stairs, move rocks, or shake hands with another tourist. To improve the user interface, users could wear special clothing, and the rover could mimic their movement. So when the user raises his hand, the rover mimics the user and raises its hand.

5.3 Going to different Environments

The rover is limited to ground movement. If we could implement the two improvements above, we could start working on a means to allow the rover go where humans can't go. An example would be sending the rover deep down into the sea, high into the sky, or even into space.

5.4 Others

The equipment that we used consisted mainly of off-the-shelf items bought from local stores. The limitations of such hardware accounted for many of the rover's shortcomings, such as high latency. Better equipment and use of a higher bandwidth connection might decrease latency and solve related problems.

CONCLUSION

Nowadays, traveling costs money and consumes time, and many people do not have the freedom to travel. Instead of enjoying the world outside their homes and workplaces, they are confined to their local environment. Fortunately, we are living in the world of technology and high speed Internet. With the help of our Internet Controlled Rover, normal people are no longer confined to their small world. No matter what their physical condition is, no matter what their financial situation is, no matter if they have time or not, they could enjoy the world outside their houses in the comfort of their houses.