

Interactive Video Management by means of an Exercise Bike

Jan Štrekelj
University of Primorska
Faculty of Mathematics, Natural Sciences
and Information Technologies
Glagoljaška 8,
6000 Koper, Slovenia
+386 5 6117 570
jan.strekelj@gmail.com

Branko Kavšek
University of Primorska
Faculty of Mathematics, Natural Sciences
and Information Technologies
Glagoljaška 8,
6000 Koper, Slovenia
+386 5 6117 654
branko.kavsek@upr.si

ABSTRACT

This paper describes the concept of virtual reality and the use of this technology in practice. The main part of the work is about reviewing the various stages of the development of a prototype system for interactive video management by means of an exercise bike. Systems, that are currently available on the market, are, due to their ease of use, closed units, for which upgrades are not possible or come at a great expense. The main advantage of the presented prototype, besides affordability, is a simple option to upgrade the system by adding sensors and/or modules; this allows us to extend the system in every stage of development. A low-cost computer (Raspberry Pi) is used as a processing unit, for calculating the speed of the wheel and sending this information to the control unit. The control unit processes the received data and sets the playback speed of the video clip accordingly. There is great potential for improvement on the developed prototype. Thus, ideas for further development are presented in the concluding section.

Categories and Subject Descriptors

H.5.1 [Multimedia Information Systems]: Augmented and virtual realities – *interactive video control*.

H.5.2 [User Interfaces]: Prototyping – *speed reading / video displaying prototype*.

General Terms

Algorithms, Management, Measurement, Performance, Design, Reliability, Experimentation, Human Factors.

Keywords

Raspberry Pi, exercise bike, virtual reality, open-source, Java.

1. INTRODUCTION

In the fast-paced life of a modern working man the time for recreation is very scarce. Not everyone can take the time to go outdoors, even going to a fitness can be either too expensive or too crowded, and hence introduces unwanted stress. Hence, purchasing a piece of indoor fitness equipment (such as an exercise bike) seem to be the perfect solution for those who cannot afford to go to a fitness regularly or have no time or will to exercise outdoors.

On the other hand, cycling on an exercise bike while staring on a wall or even watching TV and/or listening to music, can become boring after some time. This happens because the indoor environment is static and, unlike cycling on an outdoor track, offers little or no “real” motion.

It is this lack of motion and motional feedback of the exercise bike that motivated us to develop a prototype of a system that would offer the user a more realistic experience. We got the idea from the field of virtual reality, where the aim is to offer the user the illusion of reality. In our case, to make cycling on an exercise bike look more like outdoor cycling.

For this purpose, we filmed a bike ride on an outdoor track and used a bike’s speed sensor to interactively control the playback time of the video that the user is looking at.

Next section introduces the reader to the concepts of virtual reality and its use in the field of indoor exercise. In section 3 the prototype system for interactive video management by means of an exercise bike is presented in detail – the first three subsections present its architectural, hardware and software parts, while the last, fourth subsection outlines its operation. Section 4 gives conclusions and lists possible directions for further work.

2. VIRTUAL REALITY

Virtual reality is a computer technology that uses software-generated realistic images, sounds and other sensations to replicate a real environment or an imaginary setting, and simulates a user's physical presence in this environment to enable the user to interact with this space.

Let us skip the history of virtual reality and just say that virtual reality is now in its mature years. The more and more affordable prices of computer hardware and other devices (sensors, embedded systems, ...) make virtual reality appear nowadays in a wide variety of applications which include: architecture, sport, medicine, the arts, entertainment, and many more [15].

For the purpose of this paper, we focus on the application of virtual reality in sports. On websites that sell fitness equipment, we can find many indoor systems that enable interactive experience. The first such systems to appear on the market were adapted for exercise bikes. These virtual reality systems differ mainly in the way the user controls the video clip.

The first category consists of dedicated bike systems, adapted for use with virtual reality technology. Such systems are custom made and embedded in exercise bikes (in the gym) and upgrade the exercise bike with the necessary sensors that allow the user to control the video clip that is displayed on a pre-mounted display attached to the bike. An example of such a bike is shown in Figure 1.



Figure 1. A dedicated “virtual reality” indoor exercise bike (source [10]).

The second category of systems aims at bringing the cycling experience even closer to reality. This category includes devices designed for use with a real road bike. Such upgrade normally consists of two units: each one is placed under the rim of the wheel, as shown in Figure 2. The unit under the rear wheel is responsible for controlling the cycling speed and therefore the speed of video playback. At the same time, it acts as a resistor in cases where we want to simulate cycling uphill. The unit, which is mounted under the front wheel, can be reserved exclusively for ensuring the stability of the bike, the more advanced systems can record the angle of the steering wheel and, via a computer system, the image on the screen is shifted accordingly. More such bikes can be connected into the same network offering the possibility of multiplayer bike races [9] or a true VR headset can be connected to the system offering a more realistic experience [16].



Figure 2. A device under a real road bike (source [2]).

Both types of virtual reality systems for indoor cycling available on the market share the same two drawbacks: they are relatively expensive (prices ranging from several hundreds or even thousands of euro) and they are proprietary – that means they cannot be easily upgraded as needed. Our prototype system, described in detail in the next section, tries to overcome these

drawbacks by being both relatively cheap (under hundred euro) and open-source.

3. THE PROTOTYPE

This section presents in detail the prototype system and all its parts. Subsection 3.1 is dedicated to the architecture, subsections 3.2 and 3.3 describe the hardware and software part of the system respectively. Subsection 3.4 presents how all the prototype parts work together.

3.1 Architecture

The architecture of the prototype system can be divided into four parts, namely the external or input unit, the processing unit, the control unit, and the display or output unit. Figure 3 depicts the block-schema of the system, showing the connections between the units. The external unit (the exercise bike with the speed sensor attached) is connected to the processing unit via a 2-wire cable (twisted pair) and sends 0/1 signals to it. The task of the processing unit (a Raspberry Pi computer) is to transform this speed data to the actual speed of the bike’s wheel. It then sends this speed data to the processing unit (a common laptop) via the Ethernet cable using TCP/IP protocol. The processing unit is responsible for controlling the speed of the video clip that is sent to the display unit (a computer monitor) via the HDMI cable.

The four units that make up the prototype system are described in more detail in subsection 3.2 Hardware.

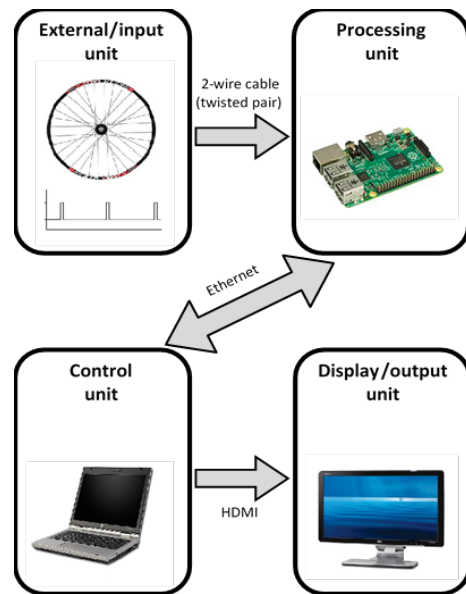


Figure 3. Block-schema of the prototype system architecture.

3.2 Hardware

In this subsection each of the four parts that make up the prototype system are presented in more detail. Thus, the following four subsections describe the external unit, processing unit, control unit, and display unit in that order.

3.2.1 External unit

The external unit (also called the input unit, because it provides the input signals to the system) is the exercise bike mounting a

speed sensor. For presentation purposes we implemented this unit as an exercise bike simulator shown in Figure 4.



Figure 4. The exercise bike simulator.

The exercise bike simulator consists of a kids bike wheel (without the tire) integrated in the forks and mounted on a pedestal (a wooden plate). The speed sensor is a magnetic switch consisting of two parts. The stationary part of the sensor is the switch itself mounted on one of the forks. The moving part of the sensor is attached on one of the spokes of the wheel and consists of a small magnet with a constant magnetic field. The rotation of the wheel causes the magnet to pass near the switch and triggers it. The detailed operation of this type of magnetic switches can be found in [3].

The magnetic switch is connected to the processing unit (via a twisted-pair cable). For safety reasons and voltage adjustment of the switch a resistance of 4.7 k Ω is connected between it and the processing unit.

The task of the external unit is to generate repetitive electric pulses and sending them to the processing unit.

3.2.2 Processing unit

For processing the signal from the external unit and calculating the speed of the wheel we used the Raspberry Pi – a low-cost, credit card sized computer [14]. The reason for using this type of computer for our processing unit is its size and versatility.

We describe here just those parts of its operation that are relevant for the understanding of our prototype system functioning (a detailed description of the Raspberry Pi can be found in [11],[14]).

The Raspberry Pi comes in many flavors. In this project the Raspberry Pi 2 Model B [11] was used. This version of the Pi comes with a 900 MHz, 32-bit Cortex-A7 CPU with 256 KB of L2 cache, a 250 MHz, VideoCore IV GPU capable of decoding 1080p video, 1 GB of RAM that is shared between CPU and GPU. Its input/output include 4 USB 2.0 slots, a CSI-2 (Camera Serial Interface, Type 2) slot, an HDMI slot, a 3.5 mm composite 4-channel analog output slot, a MicroSD card slot, a network Ethernet slot, and 40 GPIO (General Purpose Input Output) pins that can be used to connect with all types of external devices. The Raspberry Pi operates at 5V of electric charge connected to it through a MicroUSB power slot.

The external unit is connected to the processing unit by a twisted-pair cable on the processing unit's GPIO pins. Figure 5 depicts the schema of the processing unit's GPIO pins, clearly showing the pin numbering. Pins 10 and 16 were used to connect the twisted-pair cable. The "physical" connection is shown in Figure 6 in subsection 3.4.

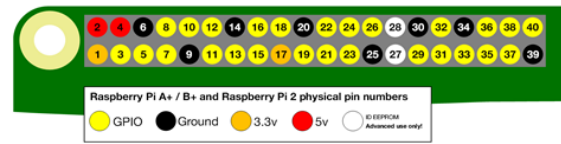


Figure 5. Raspberry Pi's GPIO schema with pin numbering (source [5]).

The task of the processing unit is to transform the electric pulses from the external unit to the actual speed of the wheel and send it to the control unit.

3.2.3 Control unit

Since the main goal of this unit is to receive the speed of the wheel from the processing unit and adjust accordingly the playback speed of the video clip to be displayed on the display unit, this unit should be efficient and stable enough to process Full HD 1080p video.

A laptop with a 2.2 GHz Intel Core 2 Duo CPU, 4 GB of RAM, and an ATI Radeon HD 4500 graphic card with 2.2 GB memory was used in our project.

3.2.4 Display unit

This unit is used to display the video clip to the user of the exercise bike. In our original idea this unit should be a computer monitor connected to the control unit by an HDMI cable, but for simplification reasons, the display of the control unit was used in the end.

3.3 Software

The external and display units, both being passive units with no processing capabilities, needed no software to run/process data. The following two subsections are thus dedicated to describe the software used to process data by the processing and control units, respectively.

Everything has been developed in the Java programming language due to its wide portability and sufficient efficiency [6]. The Eclipse programming environment has been chosen, because its ease of use and familiarity [4]. The additional libraries, specific to each unit, are presented in the following subsections.

3.3.1 Software used in the processing unit

Raspberry Pi, being the main processing unit of the system, needs an efficient operating system. For the purposes of the project we have chosen the Linux distribution Raspbian, which is a special flavor of the Debian operating system, optimized for operation on the Raspberry Pi platform [1].

To process and preparing the data from this unit for the control unit the Pi4J library of Java functions was used. Pi4J is an open source project, which aims to prepare a user-friendly environment to control the input and output of the Raspberry Pi [7]. This library provides functions for Java developers that abstracts the operation of the lower level Raspberry Pi platform specifics, and thus allowing them to focus on application development rather than to worry about underlying hardware performance [13].

The TCP/IP protocol was used to transfer the data from this unit to the control unit. Since the TCP/IP protocol implementation is part of the operating system and the knowledge about its operation is part of every computer networks course, we omit all its operational detail here – these can be found in [8].

3.3.2 Software used in the control unit

This unit is responsible for decoding the video clip to be sent to the display unit. For this purpose, VideoLAN's VLC player was used. Since video playback speed has to be dynamically adjusted to match with the data received from the processing unit, a library of functions, which offer these capabilities is needed.

Here we used the VLCj library. VLCj is an open source project by VideoLAN aimed at creating a bond between their video player VLC, and the programming language Java [12]. VLCj creates Java bindings to almost all the functionality provided by the VLC player. It greatly simplifies the development of Java applications for playing different types of multimedia, streaming video content over the Internet connection, playback mode within server applications and the implementation of video on demand. All software used for our prototype is open-source and available on <https://github.com/branax/XBike>.

3.4 Putting it all together

The final product of our project is a prototype system, which allows us to control the playback speed of a video clip with the aid of an exercise bike simulator.

The bike simulator represents the real exercise bike. Its speed sensor (in the form of a magnetic switch) is connected to a Raspberry Pi computer by a twisted-pair cable (a resistor is added to this connection for voltage adjustment purposes). Electric pulses are sent to the Raspberry Pi when the simulator's wheel rotates. The Raspberry Pi is charged by a mobile phone charger, powerful enough to provide enough electricity for its smooth functioning. The speed of the wheel is calculated by the Raspberry Pi using the received electric pulses frequency and knowing the circumference of the wheel (at sensor radius). This speed is then sent to the control unit (the laptop described in subsection 3.2.3) through the Ethernet using the TCP/IP protocol. On the laptop an instance of the VLC player is running that displays the video clip. When the player receives the speed data from the Raspberry Pi, it adjusts its playing speed accordingly. The video clip is displayed directly on the laptop's screen.

The TCP/IP connection between the processing unit (Raspberry Pi) and the control unit (laptop) is implemented in a client-server fashion, the processing unit being the server and the control unit being the client. Figure 6 shows how the finished prototype looks like.

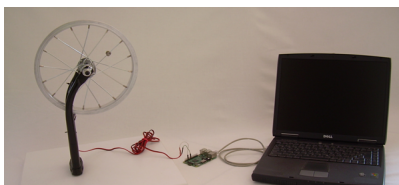


Figure 6. The prototype system.

4. CONCLUSIONS AND FURTHER WORK

Despite the fact that there are already similar products on the market, the main advantage of the presented prototype is its simple architecture and the ability to add additional modules and sensors to the processing unit. Moreover, due to its simplicity and reasonable price, our system could be accessible to different groups of users from young to old, from amateur to professional.

The modularity and the fact that our system is open source gives plenty of options for modification and/or extensions, thus offering vast possibilities for further work. The first step forward will be to implement our system on an actual exercise bike and test/evaluate its operation on real end users.

The expected increasing demand for such systems makes obvious the need for standardization of protocols, connections, modules that these systems use.

5. ACKNOWLEDGMENTS

Our thanks to Assist. Prof. Jernej Vičič. His help in the software implementation phase of the project was invaluable.

6. REFERENCES

- [1] About Raspbian, *Raspbian*. DOI= <https://www.raspbian.org/RaspbianAbout>.
- [2] BKOOL Classic Turbo Trainer. *Turbo Bike Trainer*. DOI= <http://turbobikettrainer.com/turbotrainers/bkool-turbo-trainer/>.
- [3] Campos, H. 2010. When your ICD's reed switch fails. *ICD User Group*. DOI= <http://icdusergroup.blogspot.si/2010/07/when-your-icds-reed-switch-fails.html>.
- [4] Eclipse, *Eclipse*. DOI= <http://www.eclipse.org/>.
- [5] GPIO: Models A+, B+, Raspberry Pi 2 B and raspberry Pi 3 B, *Raspberrypi*. DOI= <https://www.raspberrypi.org/documentation/usage/gpio-plus-and-raspi2/>.
- [6] Mesojedec, U., Fabjan, B, 2004. *Java 2: temelji programiranja*, Ljubljana: Pasadena.
- [7] Savage, R. Raspberry Pi with Java 8 + Pi4J. *Devovx*. DOI= <http://static1.1.sqspcdn.com/static/f/1127462/25655475/1415663597223/rpi-java-8-savage>.
- [8] Štrancar, M., Klemen, S. 2005. *PHP in My SQL na spletnem strežniku Apache*, druga dopolnjena izdaja, Ljubljana: Pasadena.
- [9] Tacx Trainers. DOI= <https://www.tacx.com/en/products/trainers/i-genius-multiplayer>.
- [10] The Coolest Bikes on No Wheels, *Espresso*. DOI= <http://espresso.com/Home>.
- [11] Turner, A. 2015. Review: Raspberry Pi 2 Model B. *PC & Tech Authority*. DOI= <http://www.pcauthority.com.au/Review/403617,review-raspberry-pi-2-model-b.aspx>.
- [12] VLCj, *Caprica Software Limited*. DOI= <http://capricasoftware.co.uk/#/projects/vlcj>.
- [13] Welcome to Pi4J!, *The Pi4J Project*. DOI= <http://pi4j.com/>.
- [14] What is Raspberry Pi, *Raspberrypi*. DOI= <https://www.raspberrypi.org/help/faqs/#introWhatIs>.
- [15] What is Virtual Reality? *Virtual Reality Society*. DOI= <http://www.vrs.org.uk/virtual-reality/what-is-virtual-reality.html>.
- [16] Widerun VR Trainer. DOI= <http://www.widerun.com/>.