Peacock: A Non-haptic 3D Performance Interface

Chikashi Miyama
University at Buffalo
431 Baird hall
Buffalo NY 14260
+1 716 868 2819
cmiyama@buffalo.edu

ABSTRACT
Peacock is a newly designed interface for improvisational performances. The interface is equipped with thirty-five proximity sensors arranged in five rows and seven columns. The sensors detect the movements of a performer's hands and arms in a three-dimensional space above them. The interface digitizes the output of the sensors into sets of high precision digital packets, and sends them to a patch running in Pd-extended with a sufficiently high bandwidth for performances with almost no computational resource consumption in Pd. The precision, speed, and efficiency of the system enable the sonification of hand gestures in realtime without the need to attach any physical devices to the performer's body. This paper traces the interface's evolution, discussing relevant technologies, hardware construction, system design, and input monitoring.

Keywords
Musical interface, Sensor technologies, Computer music, Hardware and software design

1. INTRODUCTION
Previous approaches for gesture detection of hands fall into roughly two categories: wearable devices and video analyses. An early significant achievement in the first category is Hands (1984) by Michel Waisvisz [8]. Though this sort of hand-worn device enables the detection of subtle movements of the hands or fingers, body-attached devices and/or wires leading to and from the devices could be disturbing to performers by restricting them from more agile, intensive, or dynamic actions. On the contrary, the video analysis approach, for instance, SoftVNS (2001), a set of Max objects for video analysis by David Rokeby [7], liberates performers from body-attached devices. Furthermore, this approach enables one to trace contour or color properties in captured images. However, the video analysis also has several considerable limitations. First, the system is easily affected by lights. Therefore, careful calibrations prior to performances are often required. Second, the video analysis requires sizable computational resources in order to examine a large number of pixels. Third, the frame rate of the video capture, usually 30 frames per second, is not satisfactory for realtime applications, which require higher timing precision; the perceivable latency is always evident. Fourth, since the video image is essentially two-dimensional data, the video analysis approach inevitably has difficulty in detecting three dimensional movements.

Peacock, a newly developed interface, introduced in this paper, overcomes the issues mentioned above. It offers the following five features:

1. no body-attached device
2. higher sampling rate than 30 fps
3. not affected by lighting
4. almost no CPU consumption on a host computer
5. three dimensional movement detection

2. SYSTEM DESIGN
Peacock (Figure 1) is a box-shaped interface. The size of the aluminum enclosure is 43.18 cm L x 25.4 cm W x 5.08 cm H. Thirty-five infrared proximity sensors (*GP2D12* by SHARP) are arranged in five rows and seven columns, and fixed to the back side of the top panel (Figure 2).

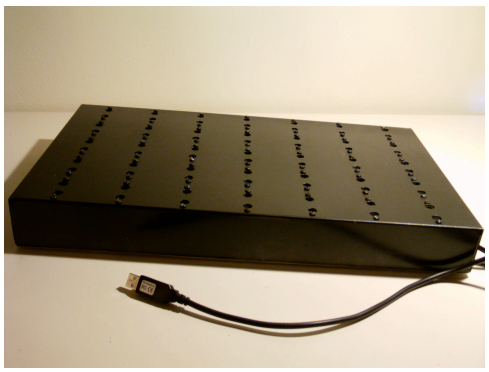

Figure 1. Peacock

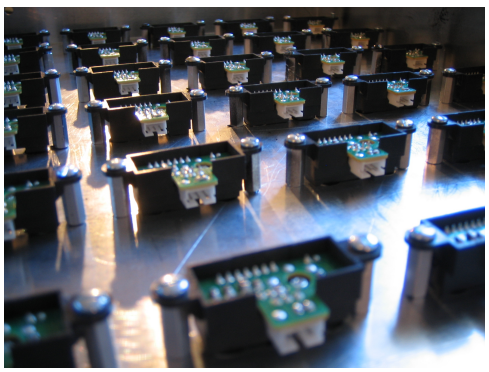

Figure 2. Infrared sensor arrays

Through the holes on the top panel, each of the sensors emits infrared rays upward and receives the reflection of the rays when an object is placed above it. The sensors vary their output voltage approximately from 0.4 to 2.6 volt based on the distance between objects and themselves within the range of 10 and 80 cm.

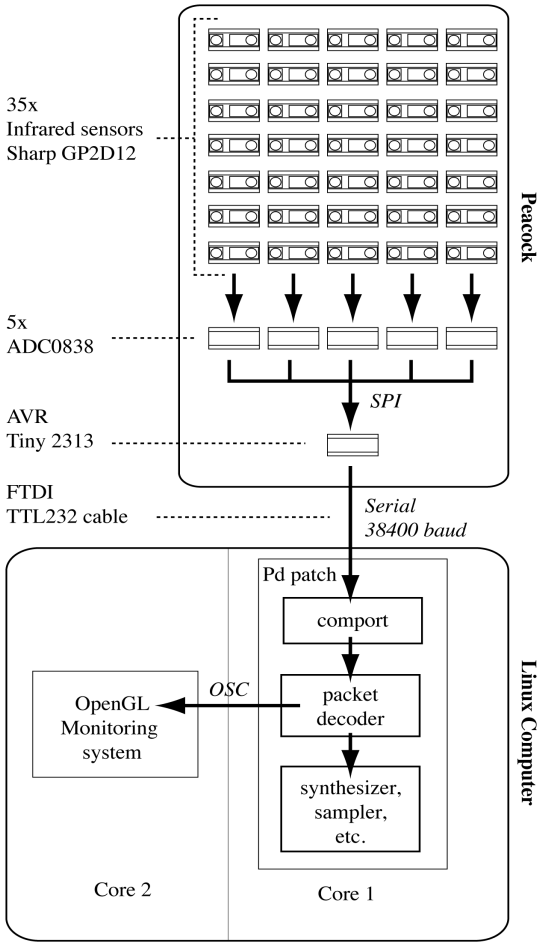

Figure 3. System design

The sensors are arranged at intervals of approximately 1.75 inches. Thus, performers can easily control two to three sensors with a palm and more than three sensors using arms.
All output from the sensors is connected to five 8 channel 8-bit ADCs (*ADC0838* by National Semiconductor [4]), with an external 3.3 volt reference voltage. They convert the voltage into 8-bit digital values and channel them to a micro-controller (*AVR ATTiny2313* by ATMEL [1]), via a four-wire SPI (Serial Peripheral Interface) bus. The micro controller packs the received data from the ADCs and sensor ID numbers into two-byte packets, and sends them to a computer using UART protocol. A TTL232 cable by FTDI [2] is utilized for transferring the UART signal over a USB cable. In order to realize this fast and efficient data transfer, the micro-controller is optimized by an external 20 Mhz crystal resonator. A Pd-extended [6] patch, running on the computer, receives the UART signal using a Pd-extended "comport" object at 38400 baud rate (Figure 3).

In the Pd patch, the received packets are unpacked to sensor ID numbers and actual 8-bit data, which can be mapped to musical parameters of various kinds.

3. GRAPHIC MONITORING SYSTEM

During the performance, a performer is able to visually monitor how sensors detect his/her gestures with an OpenGL-based [9] graphical monitoring program developed for the Linux environment. In the software, the shape of a 3D object varies based on the position of the performer's hands (Figure 4). In order to isolate audio signal processing from the monitoring processing completely, the monitoring software is developed separately from Pd using the C language, although Pd-extended offers GEM, a set of objects based on OpenGL functions. The Linux kernel later than version 2.6 provides API for CPU affinity control. This feature enables one to assign a designated thread to a specific core of a multi-core CPU [3]. This independence of processes may help avoid possible unintended interference between audio and visual processes, which might happen in a single-threaded Pd environment, and increases the stability of both audio and visual output. The Pd and OpenGL monitoring software communicate with the OpenSoundControl [10] protocol developed at CNMAT.

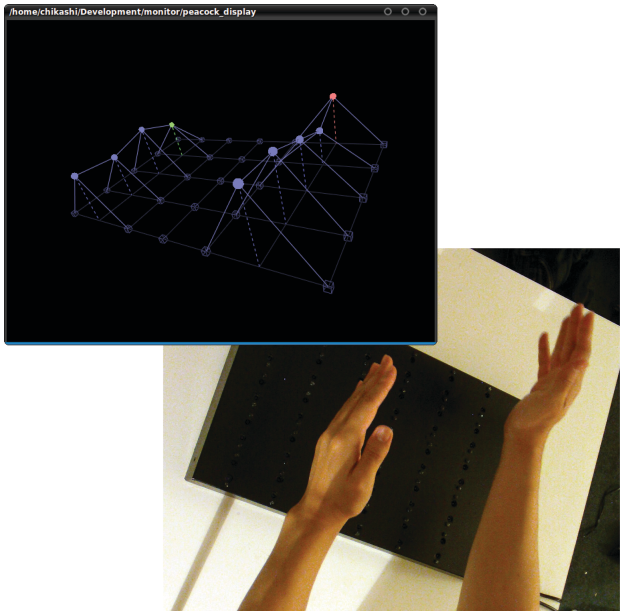

Figure 4. OpenGL monitoring software

4. EVALUATION

A benchmark patch is programmed in Pd for measuring consumed time for the analog to digital conversion and the data transfer from the ADCs to the computer via the micro-controller. The patch samples the duration of this process one thousand times, employing the "timer" object, and calculates the average.

The result of the benchmark indicates that the system requires, on average, 18.478 milliseconds for the whole process. In other words, the system works at a sampling rate close to 60 Hz, which is significantly faster—approximately twice as fast—as the frame rate of video cameras: 30 frames per second.

Moreover, the CPU load test based on Pd's "cputime" object represents lower than 1% CPU consumption for receiving and unpacking data from the micro-controller on an Intel Core 2 Duo 2.2 Ghz Macbook.

The stability of incoming voltage to the ADC was also measured by placing a sheet of white paper 20 cm above a sensor. A thousand numbers outputted from the microcontroller were recorded to an "array" object in Pd (Figure 5).

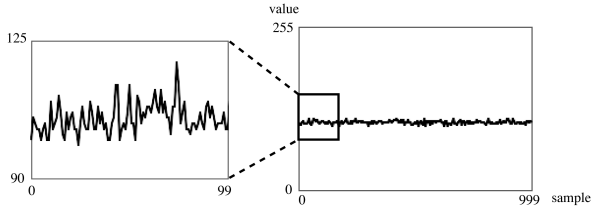

Figure 5. noise in the sampled values

The average of a thousand samples was 107.289. The minimum value was 101 and the maximum was 115. As this result shows, the noise component is not negligible. It could be reduced by applying a software lowpass filter. However, it may degrade the responsivity of the device.

5. APPLICATION

Black Vox, the first musical work for Peacock was composed by the author in December. 2009 (Figure 6). Employing the device, a performer controls more than a hundred parameters of a phase-bash-algorithm-based synthesizer [5], running in Pd-extended. The mapping between outputs from the sensors and the parameters of the synthesizer gradually varies as the piece unfolds. Thus, an identical movement of hands produces completely different musical results in different sections of the piece.

Figure 6. Performance of *Black Vox*

Moreover, in this work, one row of sensors (seven sensors) controls one voice of the synthesizer. Hence, the performer is able to control a maximum of five voices simultaneously.
This feature enables the performer to introduce further polyphonic and contrapuntal texture to the piece. The movie of the performance is available on YouTube: http://www.youtube.com/watch?v=QX1y3v3tk4w

6. FUTURE RESEARCH

As the evaluation above demonstrates, Peacock's infrared sensor-based gesture detection approach has several advantages over a hand-worn device and a video analysis approach. Future research development includedes the stabilization of sensor output by means of hardware and software noise filtering, further improvement of the OpenGL monitoring system, and the implementation of an intelligent gesture recognition system in Pd.

7. REFERENCES

[1] Atmel Corporation. AVR ATTiny2313. http://www.atmel.com/dyn/resources/prod_documents/DOC2543.PDF (accessed Dec 10, 2009)

[2] FTDI Chip. TTL-232R-USB. http://www.ftdichip.com/Documents/DataSheets/Modules/DS_TTL-232R_CABLES_V201.pdf (accessed Dec 10, 2009)

[3] GNU Operating System. Limiting execution to certain CPUs. http://www.gnu.org/s/libc/manual/html_node/CPU-Affinity.html (accessed Dec 10, 2009)

[4] National Semiconductor. ADC0838. http://cache.national.com/ds/DC/ADC0838.pdf (accessed Dec 10, 2009)

[5] Puckette, M. Phase Bashing for Sample-Based Formant Synthesis. In *Proceedings of International Computer Music Conference* (Barcelona, Spain, September 5-7, 2005), 733-736.

[6] The Pure Data Portal. http://puredata.info/ (accessed Dec 10, 2009)

[7] Rokeby, D. Soft VNS. http://homepage.mac.com/davidrokeby/softVNS.html (accessed Dec 10, 2009)

[8] Waisvisz, M. The Hands. http://www.crackle.org/The%20Hands%201984.htm (accessed Dec 10, 2009)

[9] Woo, M. Neider, J. and Davis, T. *OpenGL Programming Guide*. Addison-Wesley, 2007.

[10] Wright, M. and Freed, A. Open Sound Control: A New Protocol for Communicating with Sound Synthesizers. In *Proceedings of the International Computer Music Conference* (Thessaloniki, Greece, September 25-30, 1997), 101-104.