Low-Cost Virtual Reality System*

PlayStation 2 VR system
Technical Report No. CS03-19-00

Kuo-Chan Peter Fang
kfang@cs.uct.ac.za

Fu-Yao Kevin Feng
ffeng@cs.uct.ac.za

Ka-Man Karen Wai
kwai@cs.uct.ac.za

Supervised by:
Prof. Edwin Blake
edwin@cs.uct.ac.za

ABSTRACT
A low cost virtual reality system that generates corner projection using three PlayStation2 is presented. Two display stations each connected to one projector is used to provide panoramic view of the VR scene. A control station receives user input and broadcasts the instruction to the two display stations in order to update their respective camera positions and orientations. A demo application which immerses the user inside a glider flying through a night city has also been implemented. We report the performance of our system using random primitives. The benchmark revealed a gradual decline in frame rate in response to polygon counts in the scene. Polygon rate in our system remained near constant and does not vary with the polygon count on the screen. The results indicated for our system, a polygon count of 3540 on the screen with a refresh rate of 24fps is optimum in an interactive environment. Investigation on the relationship between roundtrip time and scene complexity revealed a significant positive correlation of (0.966). This suggests system response to user command can be delayed in a complex virtual environment.

Categories and Subject Descriptors
I.3.6 [**Computer Graphics**]: Methodology and Techniques - Interaction Techniques; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism - Virtual Reality

Keywords
PlayStation 2, Distributed, Virtual Reality

1. INTRODUCTION
While the mainstream implementations of virtual reality systems often incorporate high end hardware that are both expensive and immobile, a new emerging class of virtual reality system has gained quick acceptance in the research community. This class of virtual reality systems has been inspired by the pioneering work of Pape et al. [5] and projects like CaveQuake [6] and QuakeUT [8] which utilizes Commodity Off The Shelf (COTS) hardware in order to reduce the cost of the system down to minimum. Such an approach was previously impossible; however, consumer demand for high performance low cost graphics hardware in the gaming community has accelerated the advancement of graphics hardware in the commercial market.

The benefits of low cost virtual reality hardware are obvious; high performance systems which were previously exclusive to research institutions with well funded budget can now be constructed relatively cheaper. The reduced Price/Performance ratio has positive implications for hospitals, educational institutions, museums and other organizations where funding of new technologies are often limited. Previously disadvantaged communities can also benefit from this new technology; In education, cheap VR can provide massive quality education through the interactive learning environment; in medicine, cheap virtual environment has been shown to provide promising results in the field of exposure therapy [4].

In this paper we investigate video game console as an alternative platform to low cost distributed virtual reality (DVR) systems. By replacing high-end PCs with PlayStation2s , this project aims to establish similar VR effect at even lower cost. Another motivating factor for using PlayStation2 as a virtual reality platform is that while PCs are designed to work on a diverse range of applications, PlayStation2 is optimized only to perform high-speed graphical applications. This specialization in the hardware architecture ensures effective use of the hardware and reduces overheads associated with a general purpose architecture. The PS2 DVR System intends to provide comparable performance to high-end

DVR systems economically.

2. BACKGROUND

2.1 Previous Work

Many researchers have attempted to develop schemes to manipulate the human visual perceptual system. Cruz-Neira et al. [5] developed the first immersive virtual reality system - CAVE, at the University of Illinois at Chicago. CAVE uses multiple projectors and displays stereo on 3 walls and the floor. Rear projection is used. This approach allows the user to move freely within CAVE without obstructing the projection. Active stereo provide CAVE with the stereo capability. High-end SGI stations are used to power system. CAVE delivers amazingly immersive user experience in a real-time interactive environment.

Recent advancement in 3D acceleration cards has made low cost virtual reality viable. COTS components could be brought and assembled into a high performance virtual reality syste,. Pape et al. [6] has successfully built a passive stereo projection system using PCs equipped with high end graphics card and two projectors. Low cost VR system with similar configuration has also been built by Brouke and Bannon [11].

Yet another recent development which brings new opportunities to low cost virtual reality system is the fierce competition between video game console vendors. Game consoles with powerful graphics capabilities are now available at prices as little as R2000 per unit (current price of PlayStation2). These game consoles often outperform high end PCs in their graphics processing power. This emerging trend has brought new opportunities for VR researchers. Computational powers that were previously only possible with highly-priced super computers are now available in game consoles such as PlayStation2 and XBOX.

Specifically, the academic community has great interest in the PlayStation2 console. Although PlayStation2 only runs at 300 MHz, the machine is capable of performing over 60MFLOPS [9]. Even more importantly is the fact that PlayStation2 development is easily accessible; Sony has recently released the Linux for PlayStation2 Kit, which allows programmers to tap into the PlayStation2 hardware without purchasing the full development kit used by professional game developers. The Linux for PlayStation2 Kit converts the standard PlayStation2 into a full development environment. There are CPU and memory overheads associated with the Linux Operating System running on PlayStation2 [1].

Although there are currently no researches related to the use of PlayStation2 as a viable virtual reality platform, researchers are increasingly becoming aware of the cost benefits associated with the use of PlayStation2 hardware. Martinez [?] has demonstrated the use of PlayStation2 in computational chemistry as an alternative to PC platforms while Reed [12] is currently investigating high performance scientific visualization using clusters of 65 PlayStation2s.

2.2 PlayStation2 Architecture

The core of PlayStation2 is the Emotion Engine (EE), a 128 bit MIPS III CPU operating at 300 MHz. The CPU is supported by 32MB of RAM. Of particular importance is the two vector units embedded within Emotion Engine, VU0 and VU1. These vector units are designed to perform high speed linear transformation. The Floating Point Unit (FPU) embedded within EE is a 32 bit single precision floating point unit standard to MIPS processors. EE is the key to the high performance calculation in PlayStation2.

Another important component of the PS2 is the Graphic Synthesizer (GS), which provides 2D primitives such as lines, sprites and triangles (including triangle fans and triangle strips). GS receives instruction data in the form of Graphics Interface (GIF) packets and renders these primitives onto the screen. Data transfer in the PS2 is handled by the system wide DMA data bus. These 128 bit buses transfer data between devices within PS2 at a maximum rate of 2.4GB/sec.

Input and output is handled by the Input Output Processor (IOP) [13]. Video processing is handled by the Image Processing Unit (IPU), this device decodes MPEG2 video streams in the PS2. Stereo sound effects are processed by 2 Sound Processing Units (SPU) [3], these 2 chips when combined provides a total of 48 voice channels in a PS2 application.

Figure 1 shows how each of the above components are connected inside PS2.

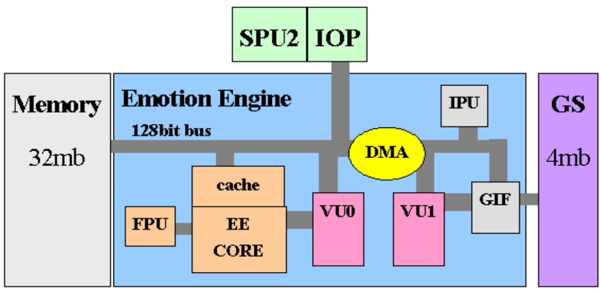

Figure 1: Basic PlayStation2 Architecture

2.3 Network Topology

The network architecture of PS2 DVR system describes how hosts are connected. The design of the distributed topology depends on the number of participants expected in the environment, the amount and the form of data being shared. In PS2 DVR system, all machines are connected in a shared centralized manner.

In shared centralized model, all shared data is stored on a central server (See Figure 2). There is no direct communication between clients. Only one client can modify the database at a time. The advantages and disadvantages of the centralized model is summarized:
Advantages:

1. Simplifies the management of multiple clients, especially in situations where strict concurrency and con-

sistency controls are required.

2. Allows server to process messages before propagating them to other clients culling, augmenting or altering the messages.

Disadvantages:

1. As an intermediary role for data delivery, additional lag can be introduced.
2. If the central server fails, none of the connected clients can interact with each other.

Despite these disadvantages, this architecture is still useful for supporting small group of collaborators. VISTEL [10] is an example that used this approach.

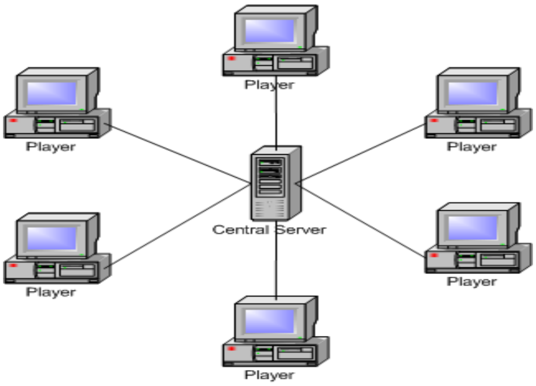

Figure 2: Shared Centralized Model

3. DESIGN

3.1 Hardware System Architecture

The PlayStation2 Virtual Reality System (Figure 3) uses front projection corner display technology to create VR experience. The system uses three identical PlayStation2 consoles to create a homogeneous network of PlayStation2. A network switch connects the three PlayStation2 via 10/100 Mbps Ethernet. The client station is responsible for gathering user input and issuing instructions to the two display servers. The standard PlayStation2 controller (DualShock2) is used to navigate through the virtual environment. The two display servers render the scene accordingly and produce output to the projectors.

3.2 Display System

Our system uses corner display; images projected from the system joins at the corner of the wall, forming a 90 degree angle between the two images. This increases the peripheral view of the user and consequently increases the user experience. Figure 5 illustrates the technique of corner display.

4. IMPLEMENTATION

4.1 Software System Architecture

The PlayStation2 Virtual Reality Display Station(Figure 5) consists of several different components which form the

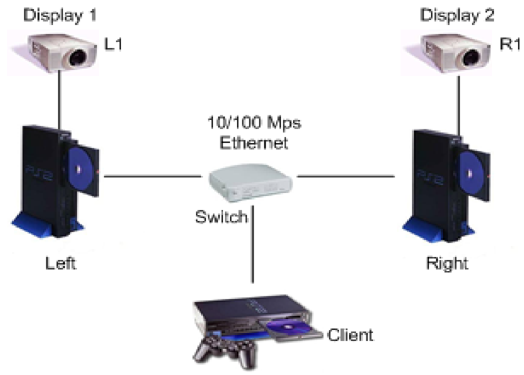

Figure 3: Diagrammatic representation of the PlayStation2 Virtual Reality System

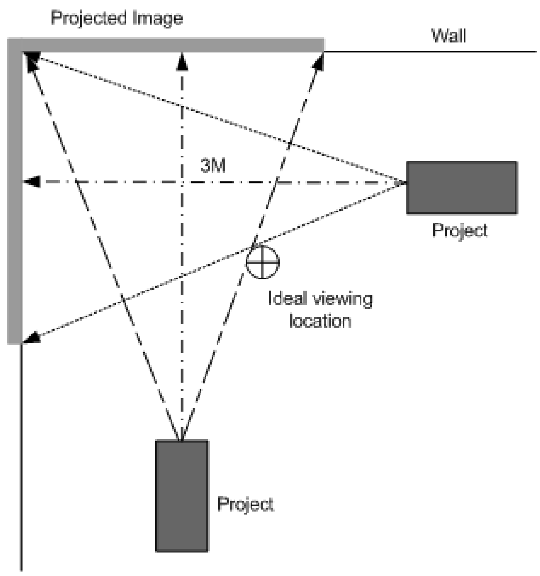

Figure 4: Diagrammatic representation of the PlayStation2 Virtual Reality System

pipeline that is shown in Figure 6. In a first step, the configuration file is read into the system. Settings in the GS are altered according to the configuration file. Our system provides 3 modes of display: VGA, NTSC and PAL. Network settings are also configured at this point. Furthermore, the system configures itself either as a left display station or right display station. Camera position and orientation is adjusted accordingly.

Figure 5: PlayStation2 VR System

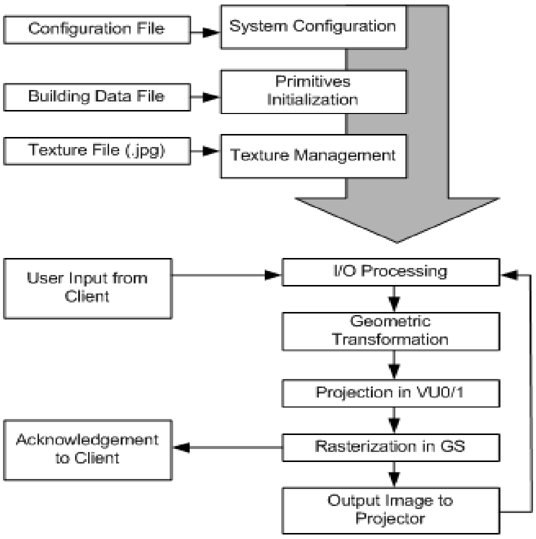

Figure 6: The system pipeline for the PlayStation2 VR display Station

The primitive initialization functions are invoked after system configuration. Primitive data are read in from the file on the system. Each of the primitive data is read into a data structure which stores the position, orientation, texture reference and texture tiling options of the primitive.

Once all the primitives have been initialized into the data structure, the texture map must be read into the memory. The texture map is converted from JPEG into suitable forms addressable by the GS. The texture tiles are each 64x64 pixels real colour. The small size of the texture tile is justified when programming PlayStation2. As the PlayStation2 is designed to display on television screens, the quality of the texture images are often unnoticeable in television resolution. Taking advantage of the situation, we were able to save memory usage and produce reasonable images when displaying using NTSC mode.

After the textures were loaded into the memory, the program enters into its main loop. User input are read from the network and processed. Camera parameters are adjusted according to user input. Geometric transformation is performed and projection is followed.

After 2D projection, data must be sent to the GS for rasterization. This is done through the construction of GIF packets. GIF packets consist of GIF tags which specify the format of the data transfer. Texture data must also be transferred at this point.

Finally, the GS interpret the information contained in the GIF packet and renders to the projector. The process is looped until user request for program termination has been received.

4.2 Network protocol

The distributed PlayStation2 VR System was implemented with User Datagram Protocol (UDP) since the system is required to work in an instantaneous environment. Furthermore, response time needs to be minimized in order to make actions coherent with the events in the virtual environment. UDP has various advantages over Transmission Control Protocol (TCP):

- It is connectionless. Thus it requires considerably less packet processing time at transmitting and receiving hosts.
- Packets can be transmitted as soon as they are sent.
- In terms of scalability, UDP is more appropriate for large-scale DVR since it does not need to maintain information for every pear hosts.

5. EXPERIMENT

Two experiments were conducted in order to evaluate the performance for our system: Hardware and Network. The hardware performance of our system is evaluated is terms of frame rate (frame/sec) and polygon rate (polygon/sec). The network performance is measured by the network roundtrip time and scene complexity (in terms of number of polygons). Figure 7 shows one of the experimental scenes used to benchmark the system.

A crucial goal in our system performance evaluation is to compare the performance of the PlayStation2 DVR System with its PC counterpart (designed concurrently with the PlayStation2 DVR System) [7].

6. BENCHMARKING

For the PlayStation2 DVR System, data is gathered by an automatic process. The required number of polygons are generated for the experimental scene, the system runs the scene and automatically displays the required statistics before program termination.

The PC DVR System [7] does not have an automatic polygon generator which conveniently generates the required num-

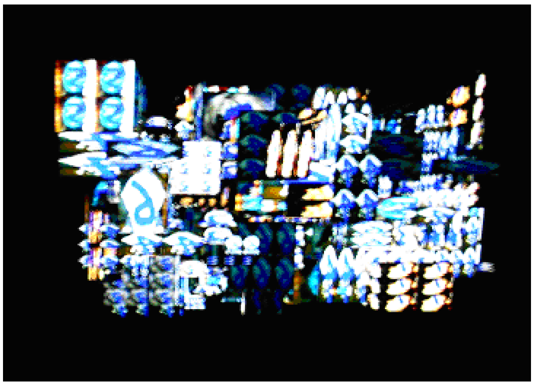

Figure 7: An experimental scene generated randomly for benchmarking

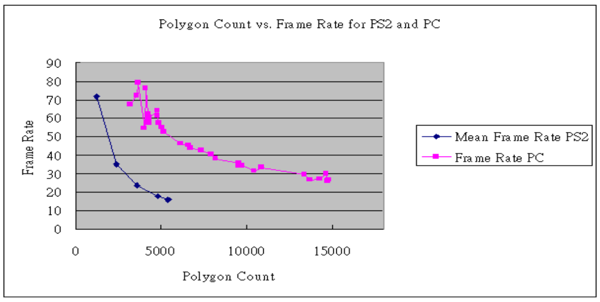

Figure 8: Frame Rate of the PS2 and PC VR Systems.

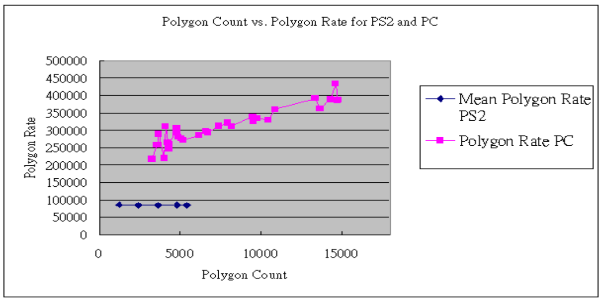

Figure 9: Polygon Rate of the PS2 and PC VR Systems.

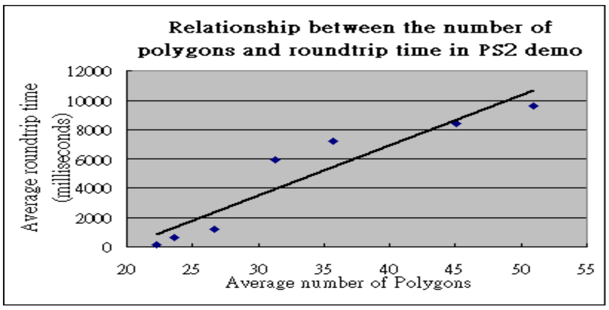

Figure 10: Graphical Representation of the mean roundtrip time and mean number of polygon of PS2 DVR System

ber of polygons for a scene. Thus, another approach is needed to gather the statistics required for our experiment. The approach which we have adopted is to randomly sample the number of polygons displayed with the frame rate and roundtrip time in the scene. As the user moves through the virtual environment in the PC DVR System, these statistics were logged into a file. A random generator is used to sample the entries in the log file.

7. RESULTS

Statistical results were calculated based on the data obtained from the experiment. The results are presented below. Table 1 summarizes the mean performance values of the PlayStation2 DVR System. Table 2 summarizes the result of the network experiment. Figure 8 and 9 compares the frame rate and polygon rate of the PS2 and PC DVR Systems.

Table 1: Mean Performance Value of the PlayStation2 VR System

Polygon Count	Mean Frame Rate	Mean Polygon Rate
1200	71.686	86022.600
2400	35.276	84662.400
3600	23.502	84605.400
4800	17.700	84957.600
5400	15.650	84507.300

Table 2: Statistical results for frame rate, number of polygons, number of primitives and roundtrip time in PS2 Demo (N = 100)

	Mean	Std Dev	Max	Min
Mean number of polygons	305.724	3993.108	332.759	10.924
Mean roundtrip time	8.442	120	10	22.285

7.1 Discussion

7.1.1 Hardware performance

As expected, the frame rate decreases as the polygon count increases. However, more importantly is the fact that PlayStation2 was able to maintain a consistent polygon rate. Analysis with Pearson Product Movement revealed an insignificant correlation between the polygon count and polygon rate (r =-0.7055, df=3, p=0.05). This suggests that the number of polygons displayed on the system per second does not deteriorate with the polygon count on the screen. By adjusting the polygon count and frame rate ratio, an optimum performance setting for the system can be attained. For our system, the average polygon rate for the vector units is 84951.060. This suggests that for an interactive frame rate of 24 frames/sec the optimum polygon count for our screen should be 3540 polygons. An optimum culling algorithm for our system should guarantee a polygon count of 3540 per frame to achieve interactive frame rate.

The PC version of the DVR system benchmarked in this experiment yields a mean polygon rate of 307822 polygons/sec. This entails that our PlayStation2 DVR System is performing at approximately 27% of the performance of its counterpart. Despite the fact that the PlayStation2 DVR System is running with the overhead from Linux, it is clear that there is room for improvement in terms of polygon rate performance.

7.1.2 Network performance

As a test of the relationship between scene complexity and roundtrip time, correlations were performed between these variables. Signaficant positive correlations were found between the variables (0.966). This is because complex scenes require extra time for memory loading of data as well as rendering, thus the server will take longer to receive an acknowledgement packet.

8. CONCLUSION AND FUTURE WORK

A clear direction for future work is to increase the system performance. One method to produce higher performance in our system is to construct DMA chain for transferring polygon data between EE and the vector units. Current implementation of our system constructs DMA packets for each polygon data. This is an inefficient method of transfer as each DMA transfer will require a DMA tag and a packet. It is more efficient to construct a long chain of DMA chain for all the polygons and sending the DMA chain in one single transfer. Performance benefit has been observed in initial investigations of this solution.

A drawback to our virtual reality solution is the restriction of user interaction to the DualShock2 controller. In a high-end virtual reality system such as the CAVE, user motion is tracked by the system and the camera corresponds to user movement. A low cost approach to provide motion tracking is to use EyeToy for PlayStation2. EyeToy is an USB device which provides user entertainment through the use of low cost video camera. The EyeToy software tracks user movement in front of the EyeToy video camera and responds with user movement. Biggar et al. [2] is currently porting the EyeToy device driver to Linux for PlayStation2.

To provide the user with a fully immersive experience, audio output can be implemented. The details of the audio system is well documented in [3] [13]. The audio devices SPS2 can be accessed in similar fashion by constructing DMA chains to SPS2. The PlayStation2 can only provide 2 channel stereo sound, however, it may be possible to simulate 5.1 Surround sound using 3 PlayStation2 , each producing 2 channels of audio, totalling 6 channels.

To summarize, a low cost virtual reality system has been built. The system developed was able to generate images at interactive rate. Our approach of replacing high end workstations with commodity video game machine has successfully reduced the overall system cost by more than 50% when compared to PC systems. Although our system performance is less than its PC counterpart, we offer an economical alternative which performed well for less demanding applications.

We envision many practical and economical applications for our new DVR system, including VR in rural education, community medicine and other scientific projects.

9. REFERENCES

[1] Play station 2 presentation linux kit.

[2] Ps2 project - eyetoy driver for playstation 2.

[3] Technical specification of playstation 2. Sony Computer Entertainment Inc.

[4] Uct exposure therapy research.

[5] T. D. C. Cruz-Neira, D.J. Sandin. Surround-screen projection-based virtual reality: The design and implementation of the cave. *ACM Computer Graphics*, 27(2):135–142, July 1993.

[6] G. D. D. Pape, J. Anstey. A low-cost projection based virtual reality display. the engineering reality of virtual reality 2002 (spie electronic imaging: Science and technology 2002). *San Jose*, January 2002.

[7] F. Feng. Low-cost virtual reality system. October 2003.

[8] J. Jacobson. Build your own pc-based cave with caveut. Department of Information Sciences at the University of Pittsburgh.

[9] T. Martinez. Computational chemistry on the playstation 2. *Martinez Research Group, University of Illinois.*

[10] K. F. T. N. Ohya J, Kitamura Y. Virtual space teleconferencing: Real-time reproduction of 3d human images. *Journal of Visual Communication and Image Representation*, 6(1):1–25, 1995.

[11] D. B. P. Bourke. A portable rear projection stereoscopic display. *Swinburne.*

[12] D. Reed. Intelligent, adaptive performance analysis; intelligent high-performance computing on toys. Pablo Research Group, University of Illinois.

[13] Sony Computer Entertainment Inc. *EE User's Manual*, 2001.