FPGA-BASED HIGH-PERFORMANCE COMPUTING FOR HAPTIC SIMULATION IN A VIRTUAL ENVIRONMENT

Rohan F. Rebello, Sriram S.
rohanrebello@gmail.com, sriramever@yahoo.co.in

Department of Computer Science and Engineering,
College of Engineering, Anna University
Chennai, India

Abstract

Touch simulation requires a refresh rate of about 1000 Hz. This has been achieved for single-point interaction. However a multiple point simulation over a sensory area requires a compromise of either surface detail or refresh rate, leading to an unsatisfactory haptic experience.

We propose an architecture using a reconfigurable FPGA co-processor to perform the application and surface-specific calculation required for rendering a surface. The process performing the rendering calculations, provides the data to the co-processor, which performs the calculation and returns the results to the process. The co-processor can be reconfigured for the respective rendering algorithm used. The use of an FPGA combines the speed of custom computing with the flexibility of reconfiguration. This enhancement does not compromise on either surface detail or refresh rate.

Index Terms

Haptics, Virtual Reality, FPGA co-processor, refresh rate, surface detail, interlacing, approximation, flexibility, multi-process, multi-threaded, threshold, reconfiguration.

1. INTRODUCTION

Touch is one of the most fundamental ways in which human beings perceive the world around them. Our understanding of the geometry and physics of the world begins by physically interacting with objects in our environment through touch. Virtual Reality (VR) has crossed the barriers of visual and aural reproduction of an environment. However to effectively simulate a completely 'real world' experience, the final frontier of touch has to be mastered. Many experiments have been conducted to this end.

It is generally accepted that, for the touch of objects in virtual environments to feel natural, the haptic information conveyed to the user must be updated in the order of 1 kHz or a thousand times per second. Computation for a single point has been performed successfully [1]. However when a multiple point interaction with the virtual environment is considered, computational complexity increases with the number of points. The result of this increased complexity would be that the system is incapable of sustaining a refresh rate of 1000 Hz for all the interaction points.

We propose the use of an FPGA co processor to accelerate the computation required by the haptic process of the virtual reality application. The FPGA unit will be used to perform computation of the collision of interaction points with objects in the virtual environment and the forces of reaction at each of the interaction points.

The general organization of the paper is as follows. Section 2 describes work related to the proposal, Section 3 details our analysis of potential solutions, our proposal and its justification. Section 4 compares this approach with an existing work followed by the conclusion in Section 5.

2. RELATED WORK

Research has been devoted to haptic interaction with objects in a virtual environment at a single point. This has been implemented in the form of the PHANToM device [2], currently available from SensAble Technologies [3]. To increase the realism of the interaction with the virtual object, a multi-point interaction interface is desirable.

Professor Mandayam A. Srinivasan and his team at MIT's Touch Lab, have developed a system using the PHANToM device, for single point haptic interaction with 3-D objects in a virtual environment. Using a Dual Pentium II, 300 MHz PC running two threads and equipped with an advanced graphics card, a haptic refresh rate of 9 to 13 kHz was achieved for the single point [1]. The architecture proposed in this paper was either multi-threaded or multi-process as the situation required, each with their own advantages. Multi-threaded architecture had the advantage of a single database of the objects that is shared between the two threads (visual and haptic). Multi-process architecture required separate databases for the two processes, but had the advantage of distributing the interaction on any desired machine.

The performance of FPGA's as pixel processors has proved to be superior to DSP's in stream oriented, signal processing type applications [4]. In such applications, a relatively small algorithm is applied to large regular blocks of data. The data moves through the logic and the results are returned to users as a stream without having to wait for the completion of the evaluation of the entire task. The requirement for a relative small algorithm implies that such algorithm must be small enough to fit into FPGA due to area and interconnect wiring constraints. The large regular blocks of data imply that potentially long execution time and opportunities for parallelization. In other words, the ratio between total execution time of a computation and overheads due to communications shall be large enough to yield high performance gain (10x-100x). This performance, although not as high as that of ASIC's, is satisfactory, and the use of FPGA's provides the added advantage of the flexibility of re-programming the co-processor to any desired algorithm.

3. OUR PROPOSAL

For a single point of interaction a refresh rate of about 10 kHz was obtained using the architecture described in [1]. If this is to be scaled for interaction at a large number of points (a human fingertip, for example has roughly 1000 individual areas of sensation [5]), the existing general purpose architecture of a processor will not be able to match the 1000 Hz refresh rate. We analyze two strategies with their drawbacks and present a solution that overcomes these drawbacks.

Analysis

To accomplish the refresh rate of 1000 Hz, required for a natural "touch and feel" of virtual objects [1], we analyze the use of two strategies: (1) Interlacing and (2) Approximation.

With the use of interlacing, a fraction (half for example) of the total points, are updated within the millisecond, and the other half are updated in the next millisecond. This reduces the number of calculations required per millisecond by half, at the expense of a detailed haptic description of the object of interaction. This is by and large satisfactory for most uniform surfaces but faces the same limitations of aliasing in graphics for more detailed textures. The end result of this would be that the user would feel a rapidly alternating description of different textures, which would seem more like vibration than an actual texture.

The main aim of approximation would be to reduce the number of interaction points, either by interpolating the existing points to new ones or by simply dropping the calculation of some points. Both these methods have the same drawback of aliasing as interlacing. Thus a more detailed surface cannot be accurately represented.

Our Solution

As can be seen in the analysis above, there is a design tradeoff between detailed representation and high refresh rate. A more detailed representation consisting

of more interaction results in a lower refresh rate, whereas a high refresh rate requires the deliberate loss of some information. Our proposal is to provide a simple solution that does not compromise on either the refresh rate or the representational detail. Our model uses an FPGA co-processor to attain the necessary refresh rate of 1000 Hz. This has the advantage of increased flexibility over ASIC's and better performance than general purpose architectures.

The co-processor whose architecture is optimized for such calculation, would speed up the calculation required. In this scenario, using either a multi-threaded or multi-process architecture to handle both the visual and haptic process [1], the co-processor could deal with the processing required for this haptic calculation, leaving the remaining processing for the main general-purpose processor. Figure 1 describes the multi-process architecture with the FPGA co-processor. The haptic interface device receives and forwards the interaction point information to the haptic process. The haptic process sends the object data from the database, and the interaction point data obtained from the haptic interface, to the co-processor which calculates the force required at each interaction point and returns it to the haptic process. This force value is then passed on to the haptic interface device which translates this into force which is actuated by the servo motors of the device. The configuration of the FPGA depends on the algorithm used to calculate the feedback force obtained from texture mapping.

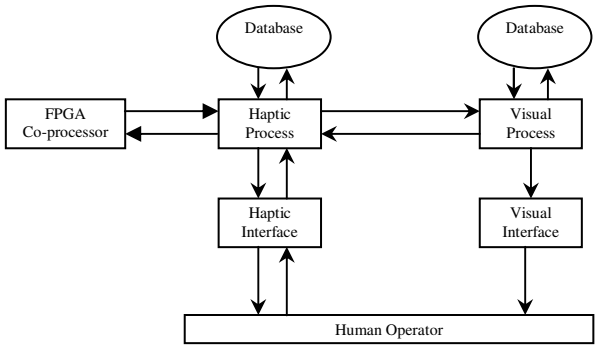

Figure 1: Multi-Process Architecture with FPGA co-processor

Justification

The use of an FPGA co-processor to speed up the calculation is justified by the improvement on execution time as well as its reconfigurability [6]. The FPGA can be configured according to the different types of surface rendering algorithms used. This flexibility is unavailable in an ASIC.

The following condition should be satisfied to justify the use of the FPGA:

$$\frac{T_h}{T_{fh}} \gg Threshold$$

where T_h is the time taken for the calculation on the host processor (in this case, a general purpose processor) without the proposed FPGA co-processor, T_{fh} is the time taken for the calculation to be executed in the proposed architecture, and *Threshold* is the speedup parameter required to obtain the desired refresh rate. T_{fh} includes the time required for host processing, FPGA processing and communication between the host and the co-processor.

Since the FPGA is configured according to the needs of the system, the execution time is expected to be lower than that of the host processor. The only bottleneck is the reconfiguration time, which is a one-time process for each algorithm used. However this bottleneck is offset by the increased performance obtained from the custom configuration. Hence the total speedup is expected to be sufficient enough to cross the threshold value.

4. COMPARISON WITH EXISTING WORK

Using a Dual Pentium II, 300 MHz PC running two threads and equipped with an advanced graphics card, a haptic refresh rate of 9 to 13 kHz was achieved for the single point [1]. Assuming linear speedup for a 3 GHz Pentium 4 PC, only about 100 such points can be maintained at the rate of 1000 Hz. For more points, some other acceleration is required. In our method this acceleration is achieved by the use of an FPGA co-processor which can be configured for the respective rendering algorithm.

5. CONCLUSION

In this paper we have outlined an architecture for reducing execution time to increase the refresh rate required for haptic simulation. Complexity is a hurdle that researchers have to overcome before a satisfactory haptic experience can be successfully obtained. We believe that our proposed architecture will be a stepping stone in this direction.

REFERENCES

[1] Ho, C.-H., C. Basdogan and M. A. Srinivasan (1999). "Efficient point-based rendering techniques for haptic display of virtual objects." Presence 8(5): 477-491.

[2] Salisbury, J. K. and M. A. Srinivasan (1997). "Phantom-Based haptic interaction with virtual objects." IEEE Computer Graphics and Applications 17(5).

[3] SensAble Technologies – The PHANTOM product line of haptic devices. http://www.sensable.com/products/phantom_ghost/phantom.asp

[4] N.W.Bergmann, J.C. Mudge, "Comparing the Performance of FPGA-based custom computer with general purpose computers for DSP applications," Proceedings of IEEE Workshop on FPGAs for Custom Computing Machines, NAPA, CA, April 1994.

[5] Robert Mittman (March 02, 2004), "Haptics—Reach Out and Touch Something", iHealth Beat, California HealthCare Foundation.

[6] Peter Mc Curry, Fearghal Morgan, Liam Kilmartin, "Xilinx FPGA Implementation of a Pixel Processor for Object Detection Applications", Communications and Signal Processing Research Unit, Department of Electronic Engineering, National University of Ireland, Galway.

Rohan Francis Rebello is currently in his final year of the B.E. course in Computer Science and Engineering at Anna University. His interests include architecture, computer networks, reconfigurable computing and human computer interaction.

Sriram Sankar is currently in his final year of the B.E. course in Computer Science and Engineering at Anna University. His interests include architecture, operating systems, reconfigurable computing and human computer interaction.