Abstract The high force update rate is a key factor for achieving high performance of haptic rendering, which imposes a stringent real-time requirement upon the execution environment of the haptic system. To fulfil this requirement, haptic systems are often limited to simplified virtual environments (VEs) for reducing computational load. This paper presents a novel pipelined threading architecture, which uses several threads running consecutively for implementing haptic rendering algorithms. With the proposed method, the haptic forces updating rate is effectively increased, which leads to improved system performance and makes complex VE scenes possible in haptic rendering. Based on the 3-DOF Delta haptic device, experiments with a virtual wall haptic system are carried out and verify the proposed method.

Keywords haptic rendering, pipelined threading, force update rate, system stability

1. Introduction

Through a haptic device (interface) and computer simulated virtual environments (VEs), haptic rendering enables human operators to feel, touch and manipulate the virtual objects. It has been shown that the ability to touch increases the immersive experience and enhances the human operator’s performance in the completion of tasks in VEs [1-3]. Haptic rendering is beneficial only if the forces are rendered to human operators with sufficient accuracy, making force fidelity an important issue in haptic rendering. For developing accurate haptic models, realistic reaction force data are usually obtained through experiments on physical objects [4, 5]. Song et al. developed a kind of multi-dimensional force/torque sensor for haptic force measurement, which can be used for verifying the accuracy of force/torque computed by haptic rendering algorithms [6]. In haptic rendering procedures, the human operator is part of the whole system, whose behaviour is hard to model. The haptic system is actually a sampled-data system [7], where the VE and the haptic device controller are discrete-time components, while the human operator and the haptic device work in continuous-time modes (Figure 1). As a result, system stability becomes a critical and fundamental issue in haptic rendering because instabilities can cause oscillations, which degrade the system performance or could even hurt the human operator. The “non-idealities”, including sample-and-
hold effect, computational delays, and asynchronous switching between continuous-time and discrete-time subsystems, are the main factors that contribute to haptic system instabilities. Research has found that computing the feedback forces at a high update rate is helpful for ensuring system stability, especially for rigid virtual objects with high stiffness [7, 8]. A force update rate of or more than 1 kHz is desired, which imposes a hard real-time requirement on the implementing of haptic systems.

Abbot et al. [10] developed a necessary and sufficient upper bound on the stiffness $K$ of the virtual wall for passivity as:

$$K \leq \min \left( \frac{2b}{2f_c T}, \frac{K}{\Delta} \right)$$  \hspace{1cm} (2)

where $f_c$ is the positive constant Coulomb friction of the haptic device and $\Delta$ is the resolution of encoder sensor for measuring position signals. Diolaiti et al. [11, 12] presented a general stability criterion for haptic system stability. They have pointed out that a haptic system with zero time delay is stable if

$$\left( \frac{b}{2f_c} \right) \left( \frac{K}{f_c} \right)^2 - 1 \geq 0 \quad \text{and} \quad f_c \geq \frac{K\Delta}{2}$$  \hspace{1cm} (3)

Where $|\phi_0|$ is the maximum velocity of the haptic device and other symbols are defined the same as above. All these equations reveal that a high force update rate is quite helpful for improving the performance of haptic systems. Gil et al. [13] studied the influence of viscous damping and time delay on the stability of haptic systems and presented the stability boundary.

Colgate et al. [14] proposed a “virtual coupling” method to limit the maximum impedance that needs to be presented by the haptic system. Hanaford and Ryu [15] defined a Passivity Observer/Passivity Controller (PO/PC) to measure energy flow in the haptic system and adaptively dissipate the excess energy to guarantee system passivity in time-domain. Gillespie and Cutkosky [16] developed controllers based on half-sample predictions and based on the design in the digital domain to eliminate “energy leaks” in the system. Ellis et al. [17] suggested a sample-estimate-hold concept and a suite of numerical methods to improve the performance of force reflections. Lu and Song [18] developed an energy-compensating control method to improve haptic system performance through calculating and eliminating “work difference” between the sampled-data haptic system and an ideal continuous-time haptic system. These works have been proved to be effective for ensuring system stabilities, but none was achieved through improving the force update rates.

Adachi et al. [9] used the intermediate representation method to maintain a high force update rate, which enables the system to display a complex environment. In their method, the reflection forces are computed using a simple virtual plane for reducing computational cost. Alhalabi et al. [19] used separate machines for decoupling haptic and graphic rendering, maximizing the independence between haptic and graphic rendering. Kirkpatrick and Sze [20] researched the influence of operating system (OS) on force update consistency under Windows 2000 and XP with single-processor and dual-processor computers. Their work mainly focused on how to guarantee a relatively high and stable force update rate.

In order to achieve high force update rates, existing haptic systems often utilize a multi-thread architecture, which decouples the haptic rendering thread from other threads such as graphic rendering [9]. However, the system still needs to apply certain simplifications to the VE or adopt relatively simple VE. The main reason is that the computational cost of a force update loop in the complex VE, consisting of collision-detection, force-response and other events, is usually too expensive for the system to render at high update rates. Obviously, this limits the realistic experience and application fields of haptic rendering. The problem gets worse as the execution environment for haptic rendering changes from laboratory development to commercial applications, where a single personal computer is usually used.

In this paper, a pipelined threading architecture is proposed to solve this problem. Several threads, starting successively with certain time intervals, are used for implementing feedback force computations, which drastically improves the equivalent force update rate. The rest of this paper is organized as follows. The related work is reviewed in Section 2. The pipelined threading architecture is presented in Section 3, followed by the experiments and results in Section 4. The concluding remarks and future research directions are given in the last section.

2. Related Work

There has been much research aimed at addressing the stability problem for haptic systems. Colgate and Schenkel [7] have derived the sufficient condition with passivity theory for the haptic rendering of a stiff wall, which relates the wall stiffness $K$ and damping $B$ with the viscous friction $b$ of the haptic device and the force update period $T$:

$$b \geq \frac{KT}{2} + B$$  \hspace{1cm} (1)
3. Pipelined Threading Architecture

With the advent of multi-core processors, multi-threading and parallel-computing technology is now emerging as mainstream, which offers a promising solution for improving the force update rate for haptic rendering. There are a variety of approaches for constructing haptic rendering applications with parallelizing programming, ranging from the use of platform-dependent threading primitives to a dedicated utility such as Intel’s Threading Building Blocks [21]. Separating the haptic rendering algorithm from other time consuming algorithms has been a standard technique for speeding up force update rate. In our method, the decoupling concept is adopted just as in other research work. However, several threads are employed for implementing the haptic rendering algorithm, which leads to a pipelined threading architecture.

![Image](image.png)

**Figure 2.** The pipelined threading architecture for haptic rendering

As illustrated in Figure 2, \( N \) haptic threads (HTs) and one visual thread are utilized to simulate the VE. Each of \( N \) haptic threads is implemented with the same update period \( T \), but starts successively in a defined time interval: HT1 starts at time \( t_0 \), HT2 starts at \( t_0 + T/N \), HT3 starts at \( t_0 + 2T/N \) and HT\( N \) starts at \( t_0 + (N-1)T/N \). With these \( N \) pipelined haptic threads, the haptic system gets an equivalent force update rate of \( N/T \), which behaves just like an ideal one with update rate \( N/T \) and with computation delay \( T \).

It is notable that all the \( N \) haptic threads share other resources in the system, i.e., the same haptic device, VE and visual rendering. The number of haptic threads \( N \) is determined by the haptic device’s maximum force update rate (including position sampling and force outputting) \( f_{device} \) and the force update rate of each haptic thread \( f=1/T \) as

\[
N = \frac{f_{device}}{f} \tag{4}
\]

4. Experiment and Results

We demonstrate the proposed pipelined threading architecture with a virtual wall prototype haptic system (Figure 3) with the Force Dimension’s 3-DOF Delta Haptic Device (DHD). A personal computer with Intel Core i5 processor and Windows 2003 operating system was utilized to display the VE. The program was written in visual C++, using the CHAI 3D open-source toolkit [22]. In our prototype haptic system, the virtual wall was displayed using the plain linear spring model with stiffness \( K \). The maximum force update rate \( f_{device} \) for the DHD was first found and then experiments were carried out to verify the proposed method.

![Image](image.png)

**Figure 3.** System configuration

The experiments were performed on five normal subjects aged 23 to 30 (two females and three males). Each subject was given a warm-up time prior to the experiments to get familiar with the haptic system and the DHD.

4.1 Maximum Force Update Rate of the Haptic Device

For DHD, its maximum force update rate \( f_{device} \) is determined by its guard time interval \( T_{guard} \), which determines the minimum time interval between two successive force outputs to prevent excessive operations. We obtained \( f_{device} \) by counting the maximum number of successful force update loops (one execution of position getting and force outputting) in some time interval. It turns out that the maximum value of \( f_{device} \) can be obtained when \( T_{guard} \) is around 200 us. Table 1 gives the maximum update rate of the DHD under different guard time intervals.

<table>
<thead>
<tr>
<th>( T_{guard}(us) )</th>
<th>Force update rate(KHz)</th>
<th>( T_{guard}(us) )</th>
<th>Force update rate(KHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>0.92</td>
<td>800</td>
<td>1.15</td>
</tr>
<tr>
<td>400</td>
<td>2.20</td>
<td>300</td>
<td>2.90</td>
</tr>
<tr>
<td>200</td>
<td>3.95</td>
<td>100</td>
<td>2.65</td>
</tr>
<tr>
<td>50</td>
<td>2.65</td>
<td>10</td>
<td>2.65</td>
</tr>
</tbody>
</table>

**Table 1.** Force update rate under different guard time intervals \( T_{guard} \) for DHD.

In the following experiments, the guard time interval \( T_{guard} \) was set to 200 us to achieve a maximum force update rate.

4.2 Maximum Achievable Stiffness

To verify the efficacy and ability of the proposed method, we tested the maximum achievable stiffness \( K_{max} \) under
different update rates and with a different number of haptic threads. As in [17], the maximum stiffness $K_{\text{max}}$ is defined as the maximum value of spring stiffness $K$ under which the human operator cannot cause the system to be unstable with any free behaviours. Considering the $f_{\text{max}}$ of the DHD is around 4.0 KHz (Table 1), the maximum number of haptic threads was set to four conservatively. For each experiment, the initial value of stiffness $K$ of virtual wall was set to 12.0 N/mm and was adjusted with step of 0.1 N/mm. The value of 20.0 N was selected to limit the maximum output forces of the DHD for safety.

All five subjects participated in the experiments and were asked to judge whether the system was stable during contact with the virtual wall under different conditions. The experiment result is given in Table 2.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Force update rate(Hz)</th>
<th>Number of haptic threads</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1(a)</td>
<td>500</td>
<td>1, 11.1, 14.4, 18.5, 20.9</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>14.9, 21.1, 24.9(b), 26.6(b)</td>
</tr>
<tr>
<td>S2</td>
<td>500</td>
<td>11.2, 15.2, 21.5</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>14.6, 20.6, 26.5, 30.7</td>
</tr>
<tr>
<td>S3</td>
<td>500</td>
<td>10.9, 14.3, 17.3, 21.7</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>14.5, 18.4, 26.9, 30.3</td>
</tr>
<tr>
<td>S4</td>
<td>500</td>
<td>10.7, 14.1, 17.6, 19.4</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>14.4, 19.1, 26.4, 32.2</td>
</tr>
<tr>
<td>S5</td>
<td>500</td>
<td>11.5, 14.9, 18.4, 21.9</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>15.2, 19.7, 27.2, 30.3</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>11.1, 14.6, 17.9, 21.1</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>14.7, 19.8, 26.4, 30.0</td>
</tr>
</tbody>
</table>

Table 2. Maximum achievable stiffness $K_{\text{max}}$ (N/mm) for virtual wall under different conditions (different force update rates and number of haptic threads). (a): Relevant data (position and force) were recorded during the experiments for Subject 1. (b): These values were taken without recording data of the experiments for avoiding response lags in visual rendering.)

Table 2 shows that the pipelined threading architecture can effectively improve the haptic system’s maximum achievable stiffness, especially under high equivalent sampling rates. Though the experiments were based on the subjects’ subjective judgments with individual differences, the improvement of $K_{\text{max}}$ was prominent enough to prove the proposed architecture. Still, due to inevitable time delays in the force update loop, the performance of the haptic system with $N$ haptic threads at force update rate $1/T$ cannot be better than that of a haptic system of real force update rate $T/N$.

Note that we initially recorded the relevant data (position and output force data of the DHD device) in HT1 at all force update rates for offline analyses in the experiments of Subject 1. However, response lags appeared in visual rendering at high equivalent update rates ($N/T$). The values in Table 2 with “(b)” superscript were obtained without the data recording operations. Perhaps the hard disk operations would affect the performance of haptic rendering at an extremely high equivalent force update rate.

4.3 Comparison Experiments

This experiment was taken to compare the performance of four haptic systems (HSs) with a virtual wall of stiffness 16.0 N/mm: HS with one haptic thread at 500 Hz update rate, HSs with two haptic threads at 500 Hz update rate, HSs with three haptic threads at 500 Hz update rate and HS with one haptic thread at 1000 Hz update rate. To reduce the influence of the hard disk operations, only the first haptic thread of each haptic system recorded the relevant data (position and force). The results for Subject 1 are shown in Figure 4.

![Image](image-url)
It is shown that HSs (Figure 4 (c)) is stable while the three other haptic systems (Figure 4 (a), (b), (d)) are unstable, which demonstrates that the pipelined threading architecture is helpful for guaranteeing the haptic system’s stability. The haptic system with several haptic threads at low update rate (500 Hz for HSs) have better performance than the haptic system with one haptic thread of a double force update rate (1000 Hz for HSs). This shows the potential of the proposed pipelined threading architecture for extending haptic systems to more complex VEs. There will be relatively sufficient time for the haptic rendering algorithms to complete the computation.

5. Conclusions and future work

5.1 Conclusions

Through using several haptic threads for computing feedback forces, the pipelined threading architecture increases the force update rates remarkably and at the same time gives relatively ample computation time for haptic rendering algorithms. This architecture has the ability to improve the performance of haptic systems (with larger maximum achievable stiffness values and improved system stability) and extend haptic rendering to more complex VEs. Experiments with a virtual wall prototype haptic system have testified and proved the proposed method. Due to its generality, the proposed architecture is applicable to haptic systems with various haptic devices and various executive environments.

5.2 Future work

Future work will investigate the combinations of the proposed pipelined threading architecture with previous methods (such as “virtual coupling” [14], PO/PC method [15] and energy-compensating control method [18]) to further improve the performance of haptic systems. The influence of timing errors for the start time of haptic threads will be studied. Detailed experiments with more subjects will be helpful for testing the virtues of the pipelined threading architecture in detail.

6. Acknowledgments

This work was supported in part by the New Faculty Start-Up Fund of Nanjing University of Aeronautics and Astronautics, by the Fundamental Research Funds for the Central Universities (NUAA Research Funding, NO. NS2012025), by Open Fund of Jiangsu Key Laboratory of Remote Measuring and Control, and by Natural Science Foundation of Jiangsu Province (Project BK2010507).

The authors gratefully acknowledge the contribution of reviewers’ comments as well as volunteers participating in our experiments.

7. References


