

Some Advances for Parallel Rendering

CHEN An-Tai¹⁺, YANG Yun-Lin², LUO Zhong-Kui²

1. School of Computer Science, Sichuan University

2. De State Key Laboratory of Fundamental Science on Synthetic Vision, Sichuan University; Chengdu 610064, China;

Abstract. In the current parallel rendering, some frame data information is lost as a result of the instability of network connection, leading to redundant controlling, inflates displaying of scenes, images stagnation and shaking, it will affect the quality of final scene display. To overcome this disadvantage, this paper puts forward mechanism of frame prestored, improving the quality of final scene display and increasing the efficiency. For each frame drawing time interval is small, the quality of scene display is better than the original. And each rendering node is independently responsible for their own rendering information in drawing this frame, reducing the redundant control and communication overhead. In addition, this paper proposes a dynamic timeout mechanism, better able to reduce time of synchronous control. Moreover, we have implemented the parallel rendering system of illumination shader-based.

Key words: frame prestored; dynamic time-out mechanism; shader-based; illumination

1. Introduction

Graphics applications have been widely applied to all kinds of fields, including scientific inquiry, engineering design, gaming, machinery manufacturing, aerospace and military, and the scale of the scene. Our imitate is becoming larger and larger, moreover the demands on the rendering speed, the rate of frame displaying, image quality and access speed are also becoming higher and higher, etc. drawing on the single computer cannot meet the actual needs [1]. Thus came into being parallel rendering based on cluster, which greatly increase the rendering speed and large-scale scene rendering accuracy and fidelity. However, there are many worthwhile improvements for the parallel rendering systems now, such as load balancing, split screen display simultaneous control and optimization of splicing and the fidelity of the natural effect. Split screen of synchronous control directly affects the display quality of the parallel rendering. The fidelity of the natural effect is also a difficulty for parallel rendering system. In this paper we present some advanced method based on the sub-screen display wall synchronization control method and illumination shader-based in the parallel rendering system to better achieve serial graphics for rendering efficiency, picture display quality and the fidelity of the natural effect.

2. Some major synchronization control method in the parallel rendering system

In the parallel rendering system, each node defines a function which is DrawSencen() to facilitate the reunification of rendering scene data, but the time each rendering nodes calls the function of DrawSencen() is usually inconsistent, in order to avoid chattering jump screen, it must synchronization control all the rendering nodes of the image output for the rendering system. And because of the instability of network traffic will result in some loss of data frame in some rendering nodes, thus increasing the redundancy

⁺ Corresponding author.
E-mail address: linyiantai@163.com.

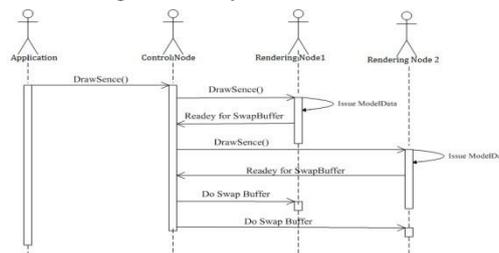
associated control, and now there are two major multi-screen display control method of synchronous parallel rendering.

2.1. The use of dedicated parallel graphics API to control frame synchronization [2] [5] [6] [7]

In this method, every parallel rendering node then has a function of DrawScene() in which set up a fence synchronous function of glBarrierExec() to achieve synchronous control, whenever there is a drawing task to complete in the rendering nodes, then glBarrierExec() will be through message passing amended accordingly, until all the rendering nodes' tasks completed, then each rendering node synchronous calls the function of glSwapBuffer() to complete simultaneous exchange of the frame buffer, finishing final rendering.

2.2. Using direct network message response to achieve the rendering nodes' frame synchronization [3] [8] [9]

In this way, the control node as the control center to keep the drawing process of all the rendering nodes synchronization, achieving all the rendering nodes' synchronous control of frame buffer.



2.3. Deficiencies for two kinds of synchronization control

- System frame rates decided by the slowest node mapping
This means that the time that the parallel rendering system to draw every frame is determined by the rendering nodes that owns the lowest frame rate.
- The heavy reliance on the network status
In addition to the efficiency of the node to rendering the drawing itself, any network delay will cause the lower frame rate of the entire system. More seriously, if the DrawScene() command cannot accepted or the Ready for SwapBuffer cannot pass back to control nodes timely because of the delay, all nodes will have to wait for the drawing, which will cause the entire drawing system standstill.

3. Slide show based on the frame prestored

As for the deficiencies of synchronization control in the rendering system above, we propose an improved synchronization control method that is the frame prestored.

3.1. Based on rendering nodes showing of synchronization control

The parallel rendering system only owns the server node and a number of rendering nodes, each frame of rendering information is processed independently in their rendering nodes after sent from the sever nodes .

3.2. Based on control nodes showing of synchronization control

This parallel rendering system consists of a server node, a number of rendering nodes and a control node, in addition to frame prestored mechanism but also set the dynamic timeout mechanism.

- The principle of the method
First of all, after the processing of the task load balanced, every frame information package will be sent to each rendering node, each rendering node then renders frame information independently, then using fence synchronization technology to ensure that all the rendering nodes display their frame rendering at the same time after a successful rendering. In order to improve efficiency and smoothness of the screen display, we set the mechanism of frame prestored in each rendering node, that is every rendering node must stores the frame information it have rendered for some time, if the frame of the next of this rendering node is rendered successful, and display the next frame, meanwhile set the next frame as the present frame and remove this frame information, so the cycle. If such a rendering node frame information is lost, as long as displaying

prestored frames in the rendering nodes. So that the final image will not appear the phenomenon of jump, fragmentary and shaking.

Meanwhile, We need to get the largest rendering time from the n-1 rendering nodes that render successfully when one rendering node' frame information lost to dynamically set the limited time of synchronization mechanism so that it can minimize rendering time of synchronous control utmost.

Assume that there are four rendering nodes which area, b, c, d, the frame rendering time of four nodes was T_1 (loss of frame' rendering time, in fact we don't know the time), T_2 , T_3 , T_4 , and $T_2 > T_3 > T_4$, then we will select T_2 as the limited time of synchronization mechanism. Assuming parallel rendering system allows the waiting time Δt , the rendering time T is:

$$T = T_2 + \Delta t,$$

If the rendering node can't display its information of frame after the time of T , then we will use the mechanism of frame prestored, and display the frame that prestored, at the same time setting this frame as the present frame.

- Processing for the loss of frame data

If there are certain packet of scenes frame data is not send to the rendering node successfully, then we will use the mechanism of frame prestored, that is getting the previous frame information instead of this frame information and setting the previous frame as the current frame.

- Dynamic time-out mechanism

In order to prevent some of the rendering node to produce problems, synchronous control of parallel rendering are all set time-out mechanism. If the frame rendering of some of rendering nodes can't post the message which means that this frame is rendered successfully to control node insides the specific time in this paper it is T we referred to above, it will enable the mechanisms of frame prestored.

Today's time-out mechanism for rendering time is static which was set before frame rendered, if all the missing frames use this time then it will affect the actual rendering time of every frame to some extent. Here we use a dynamic time-out mechanism, both stated before: in the current frame that is lost, except the missing frame of the rendering nodes, We need to get the largest rendering time from the n-1 rendering nodes that render successfully when one rendering node' frame information lost to dynamically set the limited time of synchronization mechanism so that it can minimize rendering time of synchronous control utmost.

3.3. Implementation Process

We assume the the number of rendering nodes in this render system is n , meanly there are also a server and a control node.

- Server's load balance management

The application send each frame's rendering message to the server, then the server send rendering messages to rendering nodes after load balance management.

- Synchronization control

With the received rendering messages, rendering node begins drawing. After drawing, rendering success message and rendering time would be sent to control node.

- Dynamic Time-out mechanism

After the control node received rendering success message and rendering time from the n-1 rendering node, it begins timing. If it hasn't received the message from the n rendering node over time, it enables frame prestored mechanism, and sends time-out message to that node.

- Frame prestored mechanism

If the mechanism enabled, the rendering node would set the previous frame as current frame, if a continuous time-out comes up, the system will check the network.

4. The parallel rendering system of illumination shader-based

The illumination effect based on the traditional fixed-line pipeline is not very good. This paper we have implemented the parallel rendering system of illumination shader-based.

The implementation process as follows:

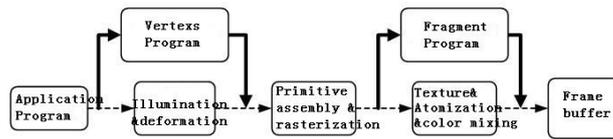
A. Vertices processing stage

In this stage, we get all the vertexes of geometry information, such as its position, its normal.

B. Fragment processing stage

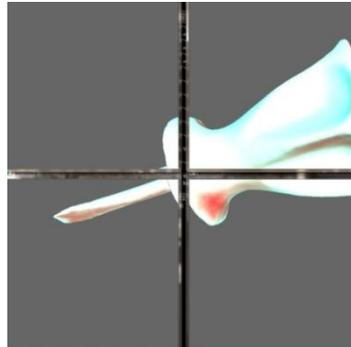
Fragment shader (fragment shader) to run in the fragment processing stage. The operation can be done at this stage are as follows:

- calculated for each pixel color, texture coordinates
- texture application
- Spray
- calculating normal

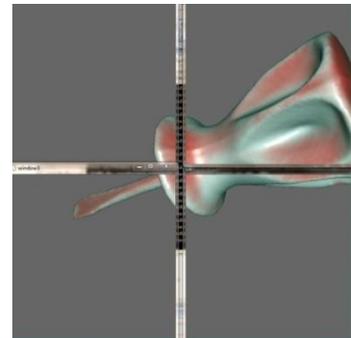


The compared of experimental result

Fixed-pipeline lighting effects as follows:



Lighting effects based-shader as follows:



The lighting effect is closer to the natural effect.

5. Experimental environment

This article uses the language of C++ to implement algorithms, the language of OpenGL to render the scene, drawing lessons from the basic equalizer parallel rendering system. The experimental system is consist of three machine with the equipment of the Intel Core 2 Duo E7400 (2.80GHz) CPU, GTX260 GPU, and 100mb / s router.

6. Summary and Prospect

The article raised a new synchronization control way to process multi-screen displaying, it made an improvement and results smoother and more lifelike scenes. Our further work is to optimize the method to gain better result.

7. Acknowledgment

This paper is supported and financed by the National High Technology Research and Development Program (863 Program) of China (Grant No. 2009AA01Z332).

8. References

- [1] Minfeng Peng. The research of Key Technology for Parallel Rendering System Architecture[D].National Defense University master's degree thesis project, 2006.6.
- [2] Wenxuan Gao. Cluster-based large-screen stereo display wall system to achieve [D]. Master of Engineering Thesis, Zhejiang University, 2006.6.
- [3] Jiaoying Shi. Parallel graphics rendering and multi-screen display Display Technology Overview[D]. Zhejiang University, CAD & CG, State Key Laboratory, 2004.8.
- [4] Thomas W.Crocket, Parallel Rendering[J].NASA Contra-actor Report 195080 ICASE Report,No.95-31.
- [5] Rudrajit Samanta, Thomas Funkhouser, Kai Li, Jaswinder, Pal Singh[2000]. Sort-First Parallel Rendering with a Cluster-ter of PCs[J]. Sketch at SIGGRAPH 2000, New Orleans, Lou-isi-ana-July 2000.
- [6] Y-chen,D.Clak,A.Finkeltein,T.House,and K.Li,Automatic Alignment of High-Resolution Multi-Projector Displays Using an Un-calibrated Camera[J].Proceedings of IEEE Visualization 2000,Salt Lake City, Utah. October2000.
- [7] H.Chen, R.Sukthankar and G.Wallace, Scalable Alignment of Large-Format Multi-Project Displays Using Camera Homography Trees[J]. Proceeding of IEEE Visualization 2002, pages 339-346, 2002.
- [8] Ruigang Yang, David Gotz, Justin Hensley, Herman Towles and Michael S.Brown, PixelFlex: A Reconfigurable Multi-Projector Display System[J]. IEEE Visualization 2001, October 2001.