Introduction

The project on Distributed Rendering Environment (DRE) builds a framework to focus research on the parallel rendering architecture and the image transmission scheme from a server to its clients. The goal of these efforts is to minimize the communication cost without impacting the image quality and load balance. Compared with the object space sorting scheme, the image space sorting scheme requires lower communication bandwidth but is prone to load imbalance. In order to overcome this disadvantage, we propose a Dynamic Mesh-based Screen Partitioning (DMSP) algorithm to partition a large task into several smaller, similar-sized tasks and assign these tasks across the rendering nodes evenly. The other challenge of DREs is the transmission of image data between the clients and the rendering engine. A novel image transmission scheme using image-based rendering techniques to solve this problem is proposed.

System Overview

The system is divided into four subsystems. They are Client Sub-system, Data Management Sub-system, Parallel Rendering Sub-system and Image Composition Sub-system.

The client system is composed of a GUI and an image assembling subsystem. The GUI provides the interaction with the user and displays the animation from the image assembling subsystem. The image assembling manager is used to compose the last image with the help from the raw image generated by the image-based rendering and the image difference from the server side.

The data management system partitions the geometry data into viewpoint cells each associated with a cull box. According to the viewpoint, it updates the primary storage memory and the secondary storage memory. It manages the data structure of the scene data to optimize performance.

There are two parts in the parallel rendering system, the controller and the rendering nodes. The controller partitions the screen space and assigns the tasks across the rendering nodes evenly. Another function of the controller is to monitor the workload of every rendering node to avoid load-imbalance. The rendering node renders a region of the screen space and sends it to the image composition system.

The image composition system receives the data from the parallel rendering system and generates an exact image. It also generates a raw image by image-based rendering technology from the old image and the predictive image. Comparing the exact image with the raw image, the comparison manager gets the difference and sends it to the client.

Parallel Rendering Algorithm

Consideration of the traditional rendering algorithm revealed that it is essentially a sorting algorithm. The taxonomy of parallel rendering was developed around 1990 by UNC (University of North Carolina) graphics researchers according to where the redistribution occurs in the rendering pipeline. It includes the classes »sort-first«, »sort-middle«, and »sort-last«.

In our system, the sort-first is chosen for its low communication cost. The sort-first is a promising
candidate for real-time distributed rendering of large datasets and high quality images. In sort-first algorithms, the image space is partitioned into non-overlapping 2D tiles and a processor does just enough geometry processing to determine the region of the raster image that a primitive will belong to. The primitive is then sent to the appropriate processor that was pre-assigned to perform both the geometry processing and rasterization for that region of the raster image. However, it is susceptible to load imbalance because of the random distribution of the primitives in the image space. In order to overcome this disadvantage, an efficient balanced screen partition is needed.

A Dynamic Mesh-based Screen Partition (DMSP) algorithm is proposed. The key idea of our approach is to cluster primitives into groups for rendering by each server dynamically based on the overlaps of their projected bounding volumes in screen space. Because of the frame-to-frame coherence, the average partition cost of every frame will not be much higher in comparison with some static partition method. This means DMSP does not need to run on every frame. It only runs when a load-imbalance event arises. A scheduler monitors the workload of each rendering node. When a load-imbalance event occurs, the scheduler launches the DMSP to repartition the screen space and reassign new jobs to the rendering nodes.

Image Transmission Scheme
Reducing the communication overhead between the remote display nodes and the rendering nodes is one of the key problems for distributed rendering systems. Typical distributed rendering systems deploy the display nodes and rendering nodes remotely. The display nodes send the camera parameters to the rendering system and the rendering system sends compressed images to the display nodes. Therefore, this solution will require high network bandwidth to display video at interactive frame rates. Temporal coherence and image space coherence are used to overcome the bandwidth limitation in our solution.

The scheme, using the camera’s motion vector, predicts the new viewpoint several frames later. The predicted frame will be rendered and sent to the client. According to the previous and predicted frame, a raw current frame is generated using image-based rendering techniques on both the client and the server. At the same time, the rendering nodes render the exact current frame, compare the raw frame and the exact frame and send only the difference to the client. This allows the server to send only incremental amounts of information for each frame, greatly reducing the bandwidth required for remote navigation.

For most cases, as shown in Figure 1, a whole image is sent once only once every 15 frames in this way; the other 14 frames are generated from image-based rendering and the difference information. Therefore, the bandwidth requirement is reduced greatly. Because of the motion prediction, the network latency can be hidden but the media generation latency is unavoidable.

Conclusions
We have described a framework for a real-time distributed rendering environment based on clusters of low-cost COTS computers. In order to overcome load-imbalance, the DMSP algorithm is proposed. A novel image transmission scheme is introduced which employs image-based rendering technology to reduce the communication cost. Future work may include the investigation of the geometry pre-processing step before sorting in detail. On the other hand, exploiting frame-to-frame coherence for sort-first parallel rendering algorithm is quite challenging and useful.

Point of contact
Zhu Huabing,
Centre for Advanced Media Technology, Singapore
Email: p149070988@ntu.edu.sg