Parallel Rendering and Display of High Resolution Realistic Images on Tiled Display Systems

專題編號:099CSIE-S025
執行期限:98年1學期至99年1學期
指導教授: 謝東儒
專題計劃參與人員: 路本庭 , 高愛琳

Abstract This paper reports the experience of building an efficient way of displaying high resolution large images on a tiled display wall using a physically based rendering ray-tracing algorithm. This system was implemented using Cross-Platform Cluster Graphic Library. The result allows us to view rendered images on a hight resolution display environment. While both tiled display systems and cluster processing are not new, there are few systems which incorporates both concepts at the same time. In the future, new systems could be developed that utilize computing clusters to render interactive ray traced images in real time.

CR Categories: I.3.7 [Computing Methodologies]: COMPUTER GRAPHICS—Three-Dimensional Graphics and Realism; I.3.2 [Computing Methodologies]: COMPUTER GRAPHICS—Graphics Systems-Distributed/network graphics.

Keywords: High-Resolution Large Display; tiled displays wall; physically based rendering system; CGLX; VideoBlaster;

1 Introduction

From movies to video games, computer-rendered images are pervasive today. We have proposed to use the physically based rendering system (PBRT) [Pharr and Humphreys 2004] based on the ray-tracing algorithm on a large tiled display wall HIPerDisplay [Hsieh 2008] for the potential of utilizing hundreds of mega pixels of contiguous display resolution.

This paper describes a system for representing high quality large images on a tiled display environment via a distributed architecture. High resolution displays have been shown to produce much greater level of effectiveness in the analysis of large data sets compared to that of a pan and zoom environment. In addition to clearer definition, the display walls also allow us to access more resource computing power at the same time. We describe how to combine these two methods to create a hybrid display / computing cluster.

2 Related Work

Creating large tiled displays [Ebara and Shibata 2009] [Ponto and Kuester 2010] has been a popular task in the scope of computer graphics. There have been many similar but fundamentally different approaches to this problem, each approach being designed for particular uses of the display.

1. Geometry Forwarding (Virtual Unified Display)

   Using DMX (Distributed Multihead X) is an optional way to display applications on a tiled display. DMX is a proxy X server that provides multi-head support for multiple displays attached to different machines (each of which is running a typical X server). DMX runs in a different mode, it distributes 2D primitives to clients instead of rendered image. This can decrease the load of both the server host and networks. By integration with Chromium, DMX can render all kinds of common GUI and OpenGL programs.

   Chromium is a system for interactive rendering on clusters of workstations. It provides sort-first, sort-last and hybrid distributed rendering ways. Most OpenGL programs can run on Chromium without modification and may not require code to be recompiled.

2. Pixel Streaming

   SAGE (Scalable Adaptive Graphics Environment) [Jeong et al. 2006] and MIND [Delgado et al. 2007] are two systems that employ different software and hardware architecture to create large high-resolution displays. In general their concepts are very similar: an application or image is transmitted over a network to a computer cluster, processed, and then eventually shown on a tiled display. Though essentially very similar, their uses and architecture are actually quite different.

   MIND allows only for single application to be rendered on the display cluster at one time employing Chromium underneath for remote parallel rendering. Chromium streams OpenGL commands to the display cluster thus reducing the amount of actual pixel data being transmitted on the network. The system is virtually transparent to the end user, one of the goals of the MIND implementation. SAGE on the other hand allows various streams to be displayed on a display cluster at one time, allowing the user to update each streams position and size on the integrated display in real time. Each stream in the SAGE system is rendered locally on the broadcaster's machine and transmitted to the remote SAGE frame buffer. The visualization software must use the SAGE Application Interface Library (SAIL) to interact with the display cluster.

3. Distributed Application

   CGLX (Cross-Platform Cluster Graphics Library) [Doerr and Kuester 2010] is another transparent framework that allows OpenGL applications to utilize high performance display clusters to maximize performance and resolution. CGLX utilizes a light-weight network communication layer that reduces many of the performance problems related to network display environments. Separate instances of the application are managed on each of the nodes in the environment. The user interacts with a GLUT-like interface which processes calls from OpenGL which distributes the data across the tiled display.

Tiled displays have also been used as a tool for displaying multidimensional data sets. Giga-Stack [Ponto et al. 2010] is one such system that utilizes massively tiled displays to render layers of multidimensional data sets. According the Giga, humans are able to process data more efficiently when it is more visually available. Thus tiled displays are more effective than single display panning / zooming techniques when trying to recognize patterns or other information in visual data sets. Giga-Stack is able to display multiple layers of data at the same time, so that the user is able to examine time-varying or multi-spectral complex records in a highly collaborative environment.

3 Implementation of PBRT on a Tiled Display Wall

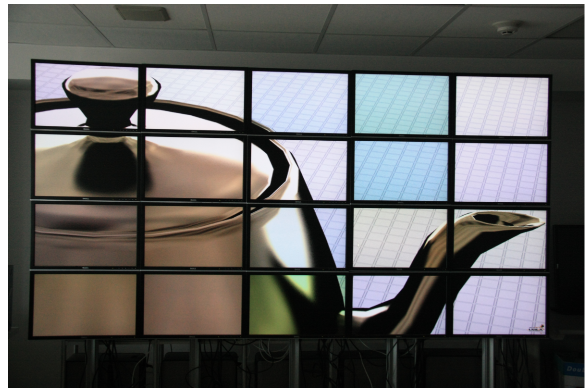
Physically Based Rendering (PBRT) is an open source cross-platform ray tracing implementation that renders photo-realistic images. PBRT by default does not provide any way to support parallel rendering across multiple machines, or scenes with animation. We have developed a rendering system based on PBRT to distribute the rendering task across a dedicated rendering cluster, as well as adding support for simple animation of lights and cameras within a scene. A single render with one frame of animation can be split into tiles across the rendering nodes. A scene with multiple frames can distributes the task by having each node render a different frame in the animation sequence. This implementation drastically speeds up rendering time. After rendering has completed, the frames are then stitched together into a single video file using the open source tool FFMpeg.

The modified PBRT incorporates a new animation description file which is read during rendering to update certain objects within the scene. The animation description files contains property "start" and "end" values which are then linearly interpolated through each frame of the animation. This allows objects within the scene, such as the camera or light attributes to be controlled more finely. The produced result is then a sequence of frames with smooth motion.

The tiled display wall in use has as a combined maximum resolution of 9600×4800 pixels. Each tile on the display has an individual resolution of 1920×1200 pixels. To match this set up, the developed PBRT generate single images of the maximum resolution. Next, the CGLX display system is used to display the produced images. CGLX automates the task of splitting the high resolution images across the multiple display devices in the system. CGLX also incorporates VideoBlaster [Ponto et al. 2009] which is capable of synchronizing the display of rendered video on each tile.

4 Conclusion

In this paper we demonstrate a method for displaying physically based rendered images on a tiled display environment. While the primary focus of this paper caters to representing static images, we have also created methods to animate the camera and light properties within the scene which are unable to be demonstrated in this paper.



References

DELGADO, J., GUILLEN, M. R., LAHLOU, M., ADJOUADI, M., BARRETO, A., AND RISHE, N. 2007. Mind: a tiled display visualization system at cate/fiu. In *GVE '07: Proceedings of the IASTED International Conference on Graphics and Visualization in Engineering*, ACTA Press, Anaheim, CA, USA, 68–73.

DOERR, K., AND KUESTER, F. 2010. CGLX: A scalable, high-performance visualization framework for networked display environments. *Visualization and Computer Graphics, IEEE Transactions on PP*, 99, 1 –1.

EBARA, Y., AND SHIBATA, Y. 2009. Study on realistic communication technology with tiled displays wall. In *NBIS '09: Proceedings of the 2009 International Conference on Network-Based Information Systems*, IEEE Computer Society, Washington, DC, USA, 612–617.

HSIEH, T.-J. 2008. Hiperdisplay (highly interactive parallelized display): A high-resolution large displays using clusters of computers. In *In Proceedings of Computer Graphics Workshop 2008*.

JEONG, B., RENAMBOT, L., JAGODIC, R., SINGH, R., AGUILERA, J., JOHNSON, A., AND LEIGH, J. 2006. High-performance dynamic graphics streaming for scalable adaptive graphics environment. In *SC '06: Proceedings of the 2006 ACM/IEEE conference on Supercomputing*, ACM, New York, NY, USA, 108.

PHARR, M., AND HUMPHREYS, G. 2004. Physically based rendering from theory to implementation. Morgan Kaufmann.

PONTO, K., AND KUESTER, F. 2010. Digi-vis: Distributed interactive geospatial information visualization. In *Aerospace Conference, 2010 IEEE*, 1 –7.

PONTO, K., WYPYCH, T., DOERR, K., YAMAOKA, S., KIMBALL, J., AND KUESTER, F. 2009. Videoblaster: A distributed, low-network bandwidth method for multimedia playback on tiled display systems. In *ISM '09: Proceedings of the 2009 11th IEEE International Symposium on Multimedia*, IEEE Computer Society, Washington, DC, USA, 201–206.

PONTO, K., DOERR, K., AND KUESTER, F. 2010. Giga-stack: A method for visualizing giga-pixel layered imagery on massively tiled displays. *Future Gener. Comput. Syst. 26* (May), 693–700.