인터랙티브 타일드 디스플레이 응용프로그램 개발을 위한 프레임워크

조용주* · 김석환**

A Framework for Constructing Interactive Tiled Display Applications

Yongjoo Cho* · Seokhwan Kim**

이 논문은 2008학년도 상명대학교 연구비를 지원받았음

요 약

본 논문에서는 인터랙티브 디지털 정보 디스플레이용 응용프로그램 개발을 도와주도록 설계된 iTDF(Interactive Tiled Display Framework)라는 타일드 디스플레이 프레임워크에 대해서 설명한다. 이 프레임워크는 고해상도 타일드 디스플레이 환경에서 여러 개의 3차원 인터랙티브 그래픽 콘텐츠를 동시에 실행시키고 사용자와의 인터랙션이 가능하도록 해준다. 이를 위해, iTDF에서는 네트워크로 묶인 여러 대의 컴퓨터를 활용해서 다수의 인터랙티브 그래픽 프로그램들을 실행시킬 수 있도록 하고, 윈도우의 위치나 크기 조정, 컴퓨터 간의 렌더링 동기화, 분산 공유 메모리, 통합 입력 인터페이스 등을 분산 시스템에서 지원한다. 본 논문은 먼저 iTDF의 요구사항에 대해서 분석하고, 그 내용을 지원할 수 있는 프레임워크의 설계와 구현에 대해서 설명한다. 그리고 기존의 데스크탑용 응용프로그램들을 iTDF를 활용해서 다시 개발하며, 유용성과 사용성에 대해 고찰한다.

ABSTRACT

This paper describes a new tiled display framework called, iTDF (Interactive Tiled Display Framework), that is designed to support rapid construction of the interactive digital 3D contents running on top of the cluster-based tiled display. This framework allows synchronizing the rendering slaves, sharing software's state over the network, the features, such as, launching multiple applications on a cluster-based computers, moving and resizing windows, synchronization of rendering slaves, distributed shared memory, and unified input interface. This paper analyzes the requirements of the framework and describes the design and implementation of the framework. A couple desktop-based applications are ported with the new iTDF and to find out the usefulness and usability of the framework.

키워드

타일드 디스플레이, 프레임워크, 인터랙션, 디지털 정보 디스플레이

* 상명대학교 디지털미디어학부(연구책임자, 교신저자)

** 상명대학교 대학원 컴퓨터과학과

접수일자 2008. 10. 07

Ⅰ. 서 론

최근에 공항, 지하철, 병원, 은행, 백화점 등과 같은 공공장소 및 일반 상점 등에 공공정보 또는 홍보물들을 보여주는 공공 정보 디스플레이가 많이 설치되어 있다. 공공 정보 디스플레이는 현재 그 적용 범위가 넓어져서 버스 정류장이나 지하철의 광고판까지도 대체하고 있을 정도로 확대되고 있다. 이런 형태가 발전하여 옥외에 각각의 목적에 맞는 디지털 콘텐츠를 제공하는 벽, 바닥, 창문, 또는 조형물로 확대될 것으로 보이며 이런 형태의 디스플레이들과 이를 개발하고 제공하는 서비스 분야가 더욱 발전할 것으로 보인다.

현재 주류를 이루고 있는 공공 정보 디스플레이는 하나의 대형 LCD/PDP를 사용하고 있지만, 최근 다수의 모니터로 구성한 타일드 디스플레이 (Tiled Display, 이하 TD) 방식이 높은 확장성 때문에 관심을 끌고 있다. 현재 공공 정보 디스플레이는 사용자 인터랙션은 제공하지 않고 주로 많은 사람들에게 동영상이나 광고사진 등의 일방적인 전달을 목적으로 만들어졌다. 그러나 앞으로는 이런 단방향성 매체보다 사용자의 요구 또는 의도를 파악하여 상황인식에 따른 서비스를 제공하는 디스플레이로 발전할 것으로 보이며, 여러 사람들이 전체 화면의 일부를 분할하여 원하는 내용을 살펴보거나 인터랙션할 수 있을 것으로 보인다.

타일드 디스플레이는 여러 개의 모니터를 단 형태(가로x세로)로 구성하여 고해상도의 대형 디스플레이를 구축할 수 있다. 타일드 디스플레이는 한 번에 많은 양의 자료를 디스플레이하여 전체 모습을 보여줄 수 있으며, 가까이 다가가면 상당히 세밀한 부분까지도 자세하게 살펴 볼 수 있게 해준다. 본 논문에서는 이러한 고해상도 타일드 디스플레이에서서 3차원 인터랙티브 콘텐츠를 자유롭게 보여줄 수 있으며, 사용자의 입력에 대해 실시간으로 반응을 보일 수 있도록 지원하는 인터랙티브 타일드 디스플레이 프레임워크(Interactive Tiled Display Framework, 이하 iTDF)를 설명한다.

iTDF 프레임워크를 사용하여, 그림 1에서 보이듯이 네트워크로 연결된 다수의 컴퓨터로 구성된 고해상도 타일드 디스플레이에서 여러 개의 실시간 3차원 인터랙티브 그래픽 응용프로그램을 자유롭게 배치시키고 실행시키며 인터랙션할 수 있도록 만들 수 있다. 본 연구에서는 다수의 인터랙티브한 콘텐츠를 타일드 디스플레이 프로그램으로 쉽게 개발할 수 있도록 iTDF에 필요한 관련 기술을 개발하였다.

기존의 TD와 관련하여 SAGE[1], Garuda[2] Chromium[3], 등과 같은 프레임워크들이 개발되었다. SAGE는 다른 컴퓨터에서 수행시킨 프로그램의 화면을 초고속 네트워크를 활용해서 스트리밍해서 보여주고 간단한 형태의 인터랙션을 지원하는 것으로 개발되었고, Garuda는 Open Scene Graph로 개발된 응용프로그램을 수정하지 않고 TD에서 보여줄 수 있도록 만들어졌다. 그리고 Chromium은 렌더링 부하를 여러 대의 컴퓨터에 나누어서 작업을 분산시키는 프레임워크로 개발되었다. 하지만 이런 프레임워크들은 주로 내용을 보여주는 것을 나누어서 그리는 작업에 목적을 두거나 기존의 화면을 그대로 스트리밍해서 보여주는 데 목적을 두었을 뿐, 여러 개의 인터랙티브 그래픽 환경을 TD에 보여주고 인터랙션이 iTDF처럼 여러 개의 인터랙티브 그래픽 환경을 TD에 직접 보여주고 인터랙션이 가능하도록 만들어진 것은 아니었다.

본 논문에서는 먼저 TD 시스템의 구성에 대해서 살펴보고, TD용 소프트웨어 개발을 위한 요구사항들을 분석한다. 그리고 프레임워크의 설계와 구현에 관해서 설명한다. 마지막으로 실제로 iTDF를 활용해서 기존의 데스크탑용 응용프로그램을 TD로 옮겼던 작업에 대해 분석하고, 추후에 진행할 연구에 대해 논한다.

Ⅱ. 인터랙티브 타일드 디스플레이 프레임워크의 설계 및 구현

2.1 타일드 디스플레이 시스템의 구성

그림 1은 본 연구에서 구축된 TD시스템을 구성한 예를 보여준다. 그림 1에서는 12개의 모니터와 7대의 컴퓨터를 활용해서 TD 시스템이 구축되었고 그 위에서 3 개의 응용 프로그램이 동작중인 것을 보여준다. 왼쪽 상단 모니터 두 개에 보인 콘텐츠는 슬레이브 컴퓨터 A에서, 왼쪽 하단은 컴퓨터 C와 E에서, 그리고 오른쪽 하단에 보인 인터랙티브 환경은 컴퓨터 A-F에 걸쳐서 실행되고 있다. 이렇게 iTDF를 이용해서 프로그램을 개발하면, 컴퓨터:프로그램의 비율을 1:1(한 개의 컴퓨터에서 한 개의 프로그램을 실행함), 1:n(한 개의 컴퓨터가 여러 개의 콘텐츠를 보여줌), n:1(여러 개의 컴퓨터가 한 개의 프로

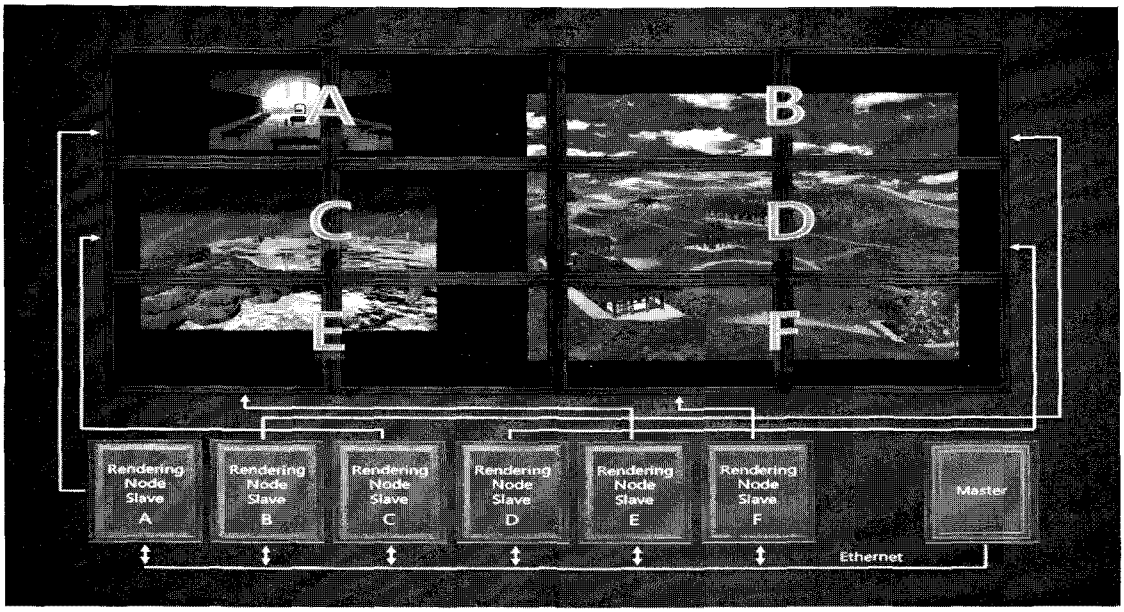

그림 1. 분산 컴퓨터를 활용하는 타일드 디스플레이 시스템의 구성 예
Fig. 1 The example of composing a tiled display based on a distributed computers

그램을 구동시키는 데 활용됨) 등과 같이 다양한 형태로 구동시키는 것이 가능하다.

그림 1에 보인 TD 시스템은 마스터/슬레이브의 구조를 갖추고 있고 각 컴퓨터들은 네트워크로 연결되어 있다. 마스터 컴퓨터는 주로 사용자의 입력 처리, 화면의 동기화, 여러 슬레이브 컴퓨터에서 프로그램들을 동시에 실행시키고 관리하기 위한 이벤트 처리 등을 담당한다. 렌더링 노드/슬레이브 컴퓨터들은 1개 또는 2개의 모니터와 연결되어 프로그램을 실행시키고 화면 렌더링을 담당하며, 마스터로부터 받은 사용자의 입력에 반응을 보이도록 설계되었다.

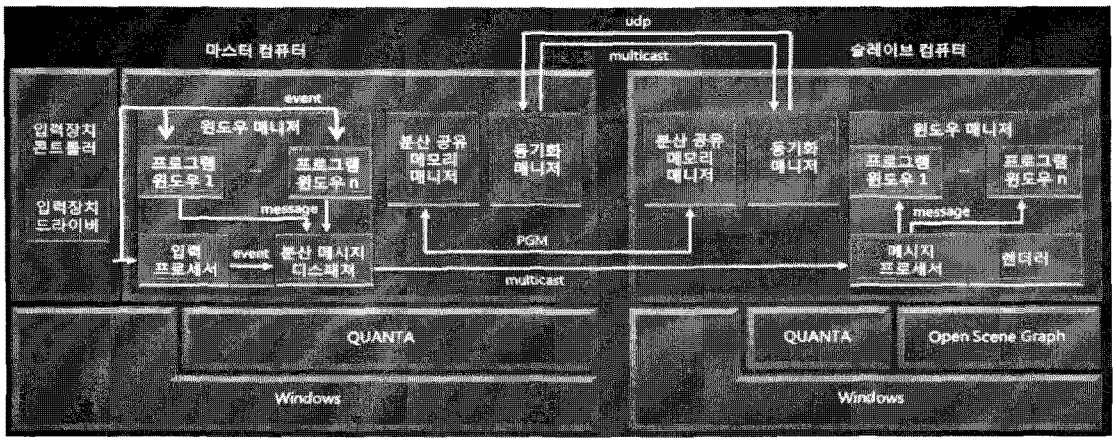

그림 2. iTDF (인터랙티브 타일드 디스플레이 프레임워크)의 시스템 구성
Fig. 2 The system architecture of iTDF (Interactive Tiled Display Framework)

2.2 프레임워크의 시스템 구조

iTDF는 TD시스템에서 사용자의 인터랙션이 지원되는 3차원 그래픽 프로그램을 쉽게 작성할 수 있도록 설계되었다. 그리고 추후에 더 높은 해상도로 시스템을 확장하려 할 때에도, 단순하게 컴퓨터와 디스플레이를 추가하고 디스플레이의 구성을 나타내는 설정 파일만 수정하면 되도록 구현하였다. iTDF는 윈도우즈 운영체제에서 개발되었고, 네트워크 처리는 윈도우즈의 Winsock2를 활용한다. 그리고 슬레이브 컴퓨터에서는 3차원 그래픽 렌더링을 위해 Open Scene Graph[4] 라이브러리를 이용한다.

마스터/슬레이브 컴퓨터들로 구성되는 타일드 디스플레이 시스템에서 동작하는 프로그램의 개발을 위해, iTDF도 그림 2에 보인 것처럼 마스터와 슬레이브용 모듈로 구성된다. 마스터 모듈은 입력장치와 연동하는 입력 모듈, 입력을 받아서 윈도우에 해당되는 이벤트와 메시지로 변환하는 윈도우 매니저, 메시지를 슬레이브 컴퓨터로 전달하는 분산 메시지 디스패쳐, 분산 컴퓨터간에 데이터를 공유할 수 있도록 해주는 분산 공유 메모리 매니저, 여러 렌더링 노드의 화면을 동시에 그림을 그릴 수 있도록 해주는 동기화 매니저 등이 있다. 슬레이브에는 각각의 프로그램 윈도우를 관리하는 윈도우 매니저, 3차원 그래픽 장면을 렌더링하는 렌더러, 마스터로부터 받은 메시지를 처리하는 메시지 프로세서 등으로 구성된다.

2.3 입력 콘트롤러와 프로세서

마스터의 입력 모듈은 모듈화 구조를 통해 다양한 입력장치들을 지원한다. 입력장치 콘트롤러는 운영체제를 통해서 입력 데이터를 추출한 후, 공유 메모리를 통해 마스터 모듈의 입력 프로세서로 전달한다. 입력장치 콘트롤러는 TD용 응용프로그램과는 별도로 만들어져서, 새로운 입력장치가 개발되더라도 기존의 응용프로그램을 수정하지 않고 콘트롤러만 바꿔서 사용할 수 있도록 하였다. 이런 기능을 지원하기 위해 iTDF에서는 입력 프로세서를 통해 전달될 수 있는 이동, 회전 등과 같은 타일드 디스플레이에서 많이 쓰이는 인터랙션에 대해 미리 이벤트 인터페이스를 정의해놓았다. 즉 입력장치 콘트롤러에서는 미리 정해진 형태로 공유 메모리에 데이터를 넣어주기만 하면, 입력 프로세서에서는 새로 업데이트 된 내용을 슬레이브 컴퓨터로 전달해서 처리시킨다.

그리고 iTDF의 입력 프로세서에서는 옵저버(Observer) 패턴[5]을 활용해서, 미리 정의되어 있지 않은 응용프로그램만의 새로운 이벤트 인터페이스를 프레임워크를 수정하지 않고도 지원할 수 있도록 하였다. 즉 응용프로그램 개발자는 iTDF에서 정의하지 않은 새로운 이벤트의 처리를 담당하는 함수를 입력 프로세서에 등록시킨다. 이렇게 등록이 되면, iTDF의 입력장치 콘트롤러로부터 새로운 형태의 이벤트가 발생할 때 바로 처리 함수로 전달해서 대응시킨다. 본 연구에서는 이렇게 새로운 입력장치 추가 기능과 새로운 인터페이스를 쉽게 등록시킬 수 있는 기능을 활용해서, 기존의 키보드와 마우스를 활용하는 입력 장치를 지원할 뿐만 아니라, Wii 게임기의 리모트 콘트롤러를 응용프로그램 수정없이도 적용시킬 수 있도록 만들었다.

2.4 윈도우 매니저와 분산 메시지 디스패쳐, 메시지 프로세서

그림 3은 iTDF를 활용한 응용프로그램이 마스터와 슬레이브 컴퓨터간에 어떻게 통신하고 동작하는 지를 보여준다. 그림에서 볼 수 있듯이 iTDF에서는 각 모듈간에 통신하고 동기화 시키는 등의 작업을 처리하기 위해 메시지를 활용한다. 역시 마스터와 슬레이브 컴퓨터들 간에서도 분산메시지 디스패쳐를 메시지를 보내서 처리하도록 구현되어 있다.

마스터와 슬레이브 모듈의 윈도우 매니저는 3차원 그래픽 환경을 보여줄 프로그램 윈도우들을 관리한다. 즉 렌더링 노드에 새로운 윈도우를 만들어서 보여주고 제거하기도 하고, 입력장치로부터 받는 데이터를 슬레이브 컴퓨터의 윈도우 매니저에 전달해서 현재 실행중인 프로그램의 위치나 크기를 바꾸는 인터랙션을 지원하기도 한다.

예를 들어, 사용자가 마우스를 이용해서 현재 실행중인 프로그램의 위치나 크기를 바꾸는 인터랙션을 취한다면, 마우스의 입력 데이터는 이벤트로 전환되어 분산 메시지 디스패쳐를 통해 슬레이브 모듈의 메시지 프로세서로 보내진다. 슬레이브 모듈에서는 받은 메시지를 분석해서 윈도우 매니저로 보내게 되고, 그런 내용들은 결국 윈도우의 위치, 크기, 또는 다른 화면 렌더링에 영향을 미치게 된다.

마스터 컴퓨터의 분산 메시지 디스패쳐(Distributed

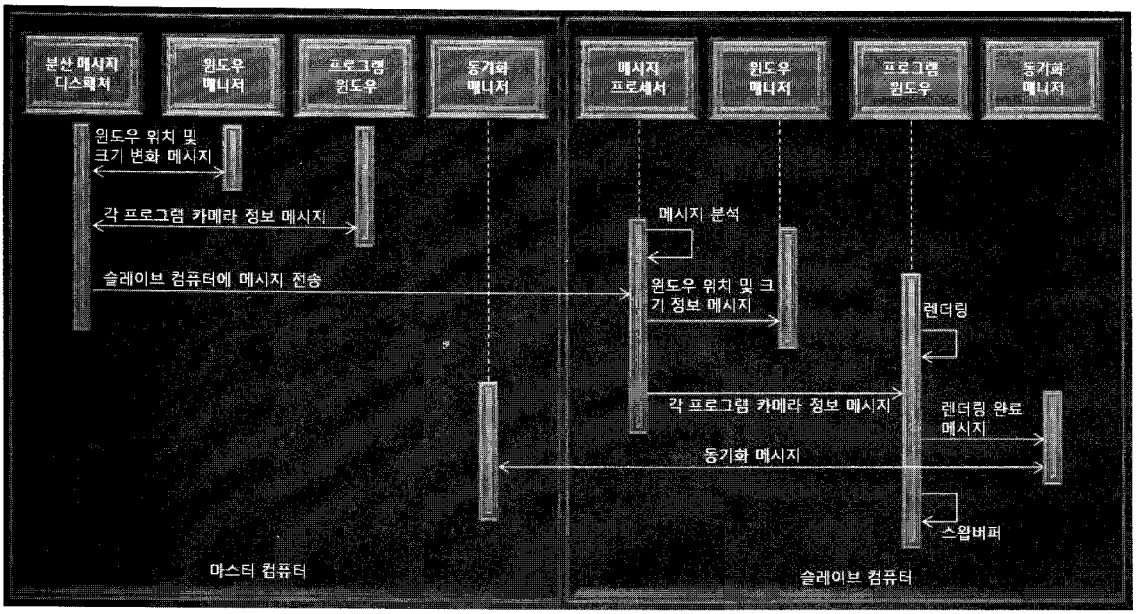

그림 3. 마스터와 슬레이브 컴퓨터간 iTDF 통신 및 동작 방법
Fig. 3 iTDF commmuncation and operation process between the master and slave computers

Message Dispatcher)는 마스터 컴퓨터에서 슬레이브의 메시지 프로세서로 데이터를 전달한다. 즉 윈도우를 움직이고 크기를 조절하는 형태의 간단한 메시지뿐만 아니라 3차원 환경에서 사용자가 보는 방향을 바꾸거나 이동하는 등과 같이 가상 카메라를 움직이는 메시지들도 슬레이브 컴퓨터의 서브 모듈에 전달하여 처리한다. 메시지를 슬레이브에서 받게 되면, 메시지 프로세서에서는 내용을 분석하여 렌더링 노드의 적절한 서브 모듈에 메시지를 전달하여 인터랙션을 수행시킨다. 분산 메시지 디스패처는 추후에 슬레이브 노드의 개수가 늘더라도 쉽게 통신할 수 있도록 만들기 위해 UDP 기반의 멀티캐스트(Multicast) 방식으로 데이터를 전송하도록 구현하였다.

2.5 동기화 매니저

화면 렌더링의 동기화는 여러 개의 프로젝션 디스플레이를 쓰는 가상현실 시스템의 연구에서도 부각되었던 문제이다. 이에 따라 가상현실시스템을 연구하는 분야에서는 SoftGenLock[6]이나, WinSGL[7]처럼 특정 하드웨어를 만들고 소프트웨어로 보완하는 방법들이 연구 개발된 적이 있다. 또한 일부 고가의 그래픽 카드에는 하드웨어적인 동기화 기능이 포함되어 있기도 하다. 하지만 이런 동기화 방법은 타일드 디스플레이 시스템을 확장시킬 때마다 새로 하드웨어를 만들어서 붙여야 하고, 또 모든 컴퓨터 하드웨어와의 호환성을 보장할 수 없다는 점과, 가격이 너무 비싸다는 단점이 있다.

그리고 iTDF시스템에서는 프로그램 윈도우의 위치나 크기가 고정되어 있던 가상현실 시스템과는 달리, 프로그램 실행 중에도 사용자의 인터랙션에 의해 그 위치나 크기가 바뀔 수 있다. 따라서 매 프레임마다 각각의 윈도우의 크기나 위치가 사용자의 인터랙션에 의해 변경되었는 지 확인하고, 바뀌었을 경우에는 슬레이브 컴퓨터에 담당해야 하는 화면의 정보를 전달한다. 이런 기능은 단순히 화면을 다시 그리는 시기만을 맞춰주는 기존의 동기화 기법으로는 구현하기가 어렵기 때문에, iTDF에서는 새로운 동기화 기법을 구현하였다.

마스터 컴퓨터에서 iTDF를 활용해서 개발된 응용프로그램이 실행되면, 마스터 모듈의 동기화매니저는 설정 파일로부터 TD시스템에서 렌더링 노드들이 어떻게 구성되었는지를 파악하고 분석한 후에, 각 슬레이브 컴퓨터에 전달한다. 그 후에 매 프레임의 장면을 렌더링할 때마다, 동기화 매니저에서는 프로그램 윈도우의 위치

및 크기 등을 고려해서 전체 3차원 공간에서의 카메라 위치 및 뷰(View)에 관한 내용들을 렌더링 노드에 전달하고, 모든 슬레이브 노드가 렌더링을 하기까지 기다린다. 각 슬레이브 컴퓨터에서는 주어진 카메라 정보 등을 활용해서 시뮬레이션 처리를 통해 장면 렌더링을 하게 되고, 프레임 버퍼를 스왑하기 직전에 마스터 컴퓨터로 스왑버퍼 준비 메시지(Ready to Swap Buffer)를 보낸다. 마스터에서는 모든 슬레이브로부터 스왑버퍼 준비 메시지를 받으면, 다시 모든 렌더링 노드들이 화면 전환을 할 수 있도록 메시지를 전달해서 화면을 갱신시킨다.

2.6 분산 공유 메모리

iTDF에서는 분산 공유 메모리(이하 DSM)라고 불리는 데이터 공유 기능을 구현하여 슬레이브 컴퓨터간에 상태 등을 공유할 수 있도록 한다. 이미 많은 연구자들이 DSM을 시스템 또는 운영체제 차원에서 또는 프로그래밍 라이브러리의 형태로 구현한 바 있다[8, 9]. 하지만 본 연구에서는 다른 연구처럼 일반화된 공유 메모리 시스템이 아니라, 간단한 형태의 데이터 공유를 필요로 하므로, 작은 양의 데이터를 공유시킬 수 있는 최소한의 기능을 갖추고, 간단하게 사용할 수 있는 경량급 분산 공유 메모리(Lightweight Distributed Shared Memory) 기법을 구현하였다.

iTDF에서의 분산 공유 메모리 기법은 메시지 기반으로 작성되었다. iTDF에서는 한 개의 슬레이브 노드가 공유 메모리에 있는 변수를 업데이트하고자 할 때 분산 공유 메모리 매니저에게 락(Lock)을 요청한다. 락이 주어지면, 변수의 값을 변경시키고 다시 언락(Unlock)시키도록 한다. 만약 공유 메모리에 있는 값이 한 컴퓨터에 의해 바뀌게 되면, 변경된 데이터는 분산 공유 메모리 매니저를 통해서 모든 슬레이브 컴퓨터들로 전달되어 각 컴퓨터에 있는 값을 변경하도록 하였다. iTDF에서는 공유 메모리의 신뢰성을 보장하면서도 확장성과 효율성을 높이는 전송방법을 채택하였다. 즉 확장성과 효율성을 높이기 위해 자주 활용되지만 데이터의 무결성을 보장할 수 없는 UDP 기반의 멀티캐스트 전송 방식이나 안정적이기는 하지만 속도가 느리고 확장성이 떨어지는 TCP 기반의 전송방식대신, 데이터 전송의 순서와 신뢰성을 보장하면서도 멀티캐스트로 전달되어 효율적인 PGM[10]을 활용하여 공유 메모리 시스템을 구현하였다.

Ⅲ. iTDF를 활용 예

그림 4는 iTDF를 활용해서 만든 가상필드와 슈퍼팡이라는 프로그램을 TD시스템에서 함께 사용하고 있는 것을 보여준다. 가상필드[11]는 몇 년 전에 초등학생들의 과학교육을 목적으로, 사방 1000미터 정도의 자갈, 모래, 그리고 잔디로 이루어진 자연적인 야외 공간을 가상현실 시스템에서 구현한 3차원 환경이다. 가상필드는 기본 환경에 꽃과 식물같은 정적인 개체와 움직이는 동물 등을 가상현실 환경 디자이너가 저작도구를 활용해서 자유롭게 배치시킬 수 있도록 만든 프로그램이다. 가상필드는 다양한 형태의 자연 현상 시뮬레이션 환경을 쉽게 꾸밀 수 있도록 설계되어서 다양한 교안을 개발하는데 적용된 바 있다.

그림 4. 가상필드와 OSG 슈퍼 팡, iTDF 활용 프로그램
Fig. 4 Virtual Field and OSG Super Pang, sample applications based on iTDF

가상필드는 가상현실 시스템에서 구현할 때에는 SGI사의 OpenGL Performer라는 그래픽 라이브러리와 Ygdrasil[12]이라는 가상환경 개발용 스크립트 언어로 만들어졌으나, 최근 OSG를 활용해서 다시 작성되었다. 본 연구에서는 OSG용 가상필드를 iTDF를 활용해서 TD 시스템으로 다시 옮겼다. 이 작업은 가상필드에서 기본 윈도우를 생성하고 렌더링하는데 활용되는 OSG 뷰어(Viewer)클래스와 사용자와의 인터랙션을 지원하는 OSG 매니퓰레이터(Manipulator)를 iTDF에서 제공하는 것으로 바꾸고, 프레임워크에서 제공하는 함수를 몇 개 추가하는 등 약 10줄 미만의 소스코드를 수정하는 것만

으로 가능하였다.

OSG 슈퍼 팡은 예전에 아케이드 게임으로 많은 인기를 끌었던 2차원 "슈퍼 팡" 게임을 OSG를 활요해서 3차원 게임으로 바꾼 것이다. 이 게임에서는 3차원 공간에 풍선이 날아다니고 사용자는 캐릭터를 조정하면서 레이저 빔을 쏴 공을 맞힌다. 풍선이 레이저 빔에 맞으면, 크기가 절반 정도인 두 개의 풍선으로 나눠진다. 그리고 사용자가 작아진 풍선을 명중시키면 또 반으로 갈라진다. 이런 식으로 사용자는 계속 풍선을 맞혀서 잘라내는 작업을 반복해서 다 제거하거나 시간이 초과될 때까지 게임을 진행하게 된다.

OSG 슈퍼 팡은 게임에서의 인터랙션을 좀 다르게 지원하기 위해 OSG 라이브러리의 소스코드를 직접 수정해서 사용했었다. 그래서 이 코드를 iTDF에서 제공하는 매니퓰레이터를 활용하도록 수정하고 코드를 추가하였다. 그리고 이 게임에서는 풍선의 위치를 분산 공유 메모리에 넣고 관리하여, 풍선이 렌더링 노드 사이를 자유롭게 움직일 수 있도록 처리하였다. 슈퍼 팡은 약 30줄 정도의 소스 코드를 수정하는 것만으로 TD 시스템으로 옮길 수 있었다. 이렇게 iTDF를 활용해서 TD 시스템용 응용프로그램을 개발하는 것은 프로그램의 복잡도에 따라 달라질 수는 있지만, 기존의 단일 데스크탑용 OSG 응용프로그램을 개발하는 것과 큰 차이가 없음을 확인할 수 있었다.

Ⅳ. 결론 및 향후 연구

디지털 정보 디스플레이 산업은 최근 시장이 급격히 커지면서 신성장 산업으로 분류될 정도로 주목받고 있다. 그리고 컴퓨팅 환경이 앞으로 유비쿼터스 컴퓨팅 환경으로 바뀌면서 대형 디스플레이들은 건물의 벽, 창문 등에도 디스플레이가 들어가는 형태로 활용도는 더욱 더 커질 것이다. 그리고 웹이 처음에는 일방적인 전달매체로 만들어졌다가, 점점 사용자가 참여하는 형태로 바뀌는 것처럼, 디지털 정보 디스플레이들도 일방적인 정보 전달에서 사용자의 참여를 요구하는 인터랙티브 디스플레이로 바뀔 것이다.

본 논문에서는 인터랙티브 디지털 디스플레이 시스템 환경 구현을 위해 TD시스템용 응용프로그램 작성을 도와주는 프레임워크를 연구하였다. 이를 위해 먼저 프레임워크의 요구사항에 대해서 분석하고, 그 내용을 지원할 수 있는 iTDF를 설계하고 구현하였다. iTDF는 인터랙티브 3차원 그래픽 환경을 개발해서 TD시스템에서 수행시킬 수 있도록 도와줄 뿐만 아니라, 그런 기능을 지원하기 위해 윈도우 매니저, 공유 메모리 관리 등의 관련 기술들을 제공한다.

iTDF를 실제로 활용해서 기존에 만들어진 OSG 응용프로그램을 타일드 디스플레이 시스템으로 옮겨본 결과, 10~30여줄 정도의 소스코드만을 수정하고도 포팅이 가능하다는 것을 알 수 있었다. 앞으로도 iTDF를 지속적으로 발전시키고 확장시켜, TD뿐만 아니라 다양한 형태의 디스플레이의 조합을 지원할 수 있도록 만들 것이다.

참고문헌

[1] Jeong, B., Jagodic, R., Renambot, L., Singh, R., Johnson, A., Leigh, A., "High-Performance Dynamic Graphics Streaming for Scalable Adaptive Graphics Environment," In Proceedings of IEEE Information Visualization Workshop, 2005.

[2] Nirnimesh, Harish, P., Narayanan, P. J., "Garuda: A Scalable Tiled Display Wall Using Commodity PCs," IEEE Transactions on Visualization and Computer Graphics, Vol. 13, No. 5, Sep/Oct, 2007.

[3] Humphreys, G., Houston, M., Ng, R., Frank, R., Ahern, S., Kirchner, P. D., Klosowski, J. T., "Chromium: a stream-processing framework for interactive rendering on clusters," ACM Transactions on Graphics, Vol 21, No. 3, pp. 693-702, 2002.

[4] OSG, http://www.openscenegraph.org

[5] Gamma, E., Helm, R., Johnson, Vissides, J. M., "Design Patterns: Elements of Reusable Object-Oriented Software," Addison-Wesley, 1995.

[6] Allard, J., Gouranton, V., Lamarque, G., Melin, E., and Raffin, B., "Softgenlock: Active Stereo and GenLock for PC Cluster," Proceedings of Joint International Immersive Projection Technologies Workshop and Eurographics Virtual Environments Workshop ((IPT/EGVE '03), 2003.

[7] Waschbüsch, M., Cotting, D., Duller, M., and Gross, M.,

"WinSGL: Software Genlocking for Cost-Effective Display Synchronization under Microsoft Windows," Proceedings of Eurographics Symposium Parallel Graphics and Visualization, 2006.

[8] Ramanjuan, R., Bonney, J., and Thurber K., "Network shared memory: a new approach for clustering workstations for parallel processing," Proceedings of the IEEE International Symposium on High Performance Distributed Computing, pp. 48-57, 1995.

[9] 이상권, 윤희철, 이준원, 맹승렬, "KDSM(KAIST Distributed Shared Memory) 시스템의 설계 및 구현", 한국정보과학회 논문지, Vol. 29, No. 5, pp. 257-264, 2002.

[10] Jones, A. and Ohlund, J., "Network Programming for Microsoft Windows," MS Press, pp. 327-334, 2002.

[11] 조용주, 박경신, "가상필드에서 초등학생들의 과학적 탐구를 도와주는 멀티미디어 보조도구," 한국정보처리학회논문지, pp. 143-150, 4월, 2005.

[12] Pape, D., "Composing Networked Virtual Environments," Ph.D. Dissertation, University of Illinois at Chicago, 2001.

저자 소개

조용주 (Yongjoo Cho)

1993년 University of Illinois at Urbana-Champaign 컴퓨터과학과 (학사)
1997년 University of Illinois at Chicago 전기전자컴퓨터과학과(석사)
2003년 University of Illinois at Chicago 컴퓨터과학과 공학(박사)
2004년 ~ 현재 상명대학교 디지털미디어학부 조교수
※관심분야 : 가상현실, 인터랙티브 컴퓨팅, 에듀테인먼트

김석환 (Seokhwan Kim)

2007년 상명대학교 소프트웨어학부 졸업 (학사)
2007년~현재 상명대학교 대학원 컴퓨터과학과(석사과정)
※관심분야 : 타일드 디스플레이 시스템, 모바일 웹서비스