

가

김기범¹, 김대현², 박성원³, 김명준⁴
그래픽스연구원¹²³, 이화여자대학교 디지털미디어학부⁴
{gbkim¹, dhkim², swpark³}@igi.re.kr, {mjkim⁴}@ewha.ac.kr

Construction of Low Cost Tiled Display System with Super High Resolution

Gi Beom Kim¹, Dae Hyun Kim², Seong Won Park³, Myoung-Jun Kim⁴
Institute for Graphic Interfaces¹²³, Ewha Womans University⁴

요약

본 논문에서는 저가의 보급형 프로젝터를 사용하고 Programmable GPU 기법중 Pixel Shader 기술을 이용하여 에지블렌딩을 수행하였으며, 7×4 개의 프로젝터로 구성된 6592×2784 픽셀의 초고해상도를 가지는 5.6m×2.4m의 대형 타일드 디스플레이를 구축해 보았다. 또한 타일드 디스플레이용 응용 프로그램으로서 타일드 디스플레이 시스템을 마치 하나의 컴퓨터처럼 작동시킬 수 있는 타일드 디스플레이 관리 프로그램을 개발했으며, 이 프로그램은 컴퓨터와 프로젝터 제어, 응용프로그램 실행 및 종료를 담당한다. 그 외에도 일반 컴퓨터에서는 실행이 불가능한 초고해상도의 이미지 및 동영상까지도 볼 수 있는 이미지 뷰어와 동영상 플레이어 개발하였다. 또한 100 만 폴리곤 이상의 3D 모델을 실시간으로 인터랙션 할 수 있는 3D 뷰어 등을 개발 하였다.

Keyword : Tiled Display, Multi-Projection, Virtual Reality

1. 서론

데이터 가시화와 같은 과학분야와 상황 파악을 위한 통제실, 방송/엔터테인먼트, 교육 등 다양한 부문에서 고해상도 대형 디스플레이의 요구가 증가되고 있다. 이런 초고해상도의 대형 디스플레이는 사용자들에게 실물과 같은 화면의 정보를 공유하게 해줌으로써 좀더 현장감을 증대시켜준다.

현재 이런 초고해상도 대형 디스플레이의 종류로는 일반 프로젝터보다 높은 해상도와 밝기를 가지는 고가의 프로젝터를 스크린에 투영하는 방식과 <그림 1>과 같이 LED 램프를 가로/세로로 배열한 디스플레이 방식과 <그림 2>와 같이 대형 LCD/PDP 디스플레이를 사용하는 방식, <그림 3>과 같이 여러 대의 LCD/PDP 을 연결하여, 마치 하나의 디스플레이 처럼 보이도록 하는 LCD/PDP 형 타일드 디스

플레이 방식, <그림 4>와 같이 여러대의 프로젝터들을 이용하여 스크린의 각 부분을 투영함으로써 하나의 디스플레이처럼 보이도록 하는 프로젝터형 타일드 디스플레이 방식이 있다.



<그림 1> LED 를 사용한 디스플레이



<그림 2> PDP 디스플레이



<그림 3> LCD 형 타일드 디스플레이



<그림 4> NCSA의 프로젝터형 타일드 디스플레이

고해상도 프로젝터를 사용하는 경우는 현재 시판되는 프로젝터의 최대 해상도가 2048×1080 픽셀이어서 큰 화면을 구성하기 위해서는 스크린과 프로젝터간의 투영거리가 멀어져야 한다. 이로 인해 화면은 커지지만, 해상도가 떨어지게 되고 단위 면적당 화면의 밝기가 감소하여 화면이 전체적으로 어두워지게 된다. <그림 1>과 같이 LED 를 사용한 디스플레이는 RGB 색상을 낼 수 있는 LED 들이 가로, 세로로 디스플레이 판에 촘촘히 박혀서 모니터의 픽셀과 같은 역할을 수행하게 된다, 그러나 단위 LED 의 크기로 인해 가까이에서 영상을 보게 되면 가로, 세로 줄무늬가 나타나므로 스크린과 일정 간격을 두고 보아야 한다. <그림 2>와 같이 LCD/PDP 디스플레이는 현재 최대 해상도가 1920×1080 픽셀이고 크기 또한 2.31m×1.325m 까지만 개발된

상태이다. <그림 3>과 같은 LCD/PDP 형 타일 디스플레이는 각각의 LCD/PDP 액정을 타일처럼 배열하므로 타일간 이음이 선명하게 보이게 된다. <그림 4>와 같은 프로젝터형 타일드 디스플레이는 각각의 프로젝터들을 격자형으로 배열하면 초고해상도 화면을 구성할 수 있고 프로젝터와 스크린의 간격도 훨씬 가까워져서 더욱 밝은 화면을 구성할 수 있다.

그러나 타일드 디스플레이 시스템이 하나의 화면처럼 보이기 위해서는 이웃하는 프로젝터간의 칼라의 차이를 느낄 수가 없어야 한다.

본 논문에서는 저가의 보급형 프로젝터를 사용하며 Programmable GPU 기법중 Pixel Shader 기술[1]을 이용하여 에지블렌딩을 수행하였으며, 7×4 개의 프로젝터로 구성된 6592×2784 픽셀의 초고해상도를 가지는 5.6m×2.4m 의 대형 타일드 디스플레이를 구축해 보았다. 타일드 디스플레이용 응용 프로그램으로서 타일드 디스플레이 시스템을 마치 하나의 컴퓨터처럼 작동 시킬 수 있는 타일드 디스플레이 관리 프로그램을 개발하였다. 이 프로그램은 컴퓨터와 프로젝터 제어, 응용프로그램 실행 및 종료를 담당한다. 그 외에도 일반 컴퓨터에서는 실행이 불가능한 초고해상도의 이미지 및 동영상까지도 볼 수 있는 이미지 뷰어와 동영상 플레이어를 개발하였다. 또한 100 만 폴리곤 이상의 3D 모델을 실시간으로 인터랙션 할 수 있는 3D 뷰어 등을 개발 하였다.

본 논문의 구성은 다음과 같다. 2 장에서는 기존의 타일드 디스플레이 구축 사례를 살펴보고 3 장과 4 장에서는 타일드 디스플레이를 구축할 때 사용된 하드웨어 기법과 구축된 타일드 디스플레이 응용 어플리케이션에 대해서 살펴보고 마지막 5 장에서는 결론 및 향후 과제에 대하여 기술한다.

2. 기존의 연구

국내의 연구 사례로는 한국과학기술정보연구원(KISTI)에서 타일드 디스플레이를 제어하는 1 개의 master 컴퓨터과 렌더링을 위한 6 개 slave 컴퓨터를 사용하여 3×2 의 19" LCD

모니터로 구성된 리눅스 기반의 타일드 디스플레이를 구축하였으며[2][3], 각각의 LCD 해상도는 1280×1024 픽셀이며 이 타일드 디스플레이 소프트웨어로써는 컴퓨터에 연결된 LCD 모니터들을 하나로 묶어서 단일 데스크톱 화면처럼 사용하는 X 서버인 DMX(Distributed Multithread X)[4]를 사용하였으며, 클러스터에서 애플리케이션 실행 및 클러스터 관리를 위한 툴인 RSH[5]를 사용하였다. 이 타일드 디스플레이는 주로 IRIX 가시화 전용 소프트웨어를 사용하여 계산 데이터 가시화용으로 사용하였으며, 각 slave 컴퓨터간의 동기화 및 렌더링을 위해 CAVELib 를 사용함으로써 6 개 화면까지만 확장이 가능한 제약이 있다.

해외의 연구 사례로, 미국 일리노이스 대학 EVL(Electronic Visualization Laboratory)의 LambdaVision[6] 은 <그림 5>과 같이 11×5 의 LCD 형 타일드 디스플레이를 구현하였다. 각각의 LCD 해상도는 1600×1200 픽셀의 해상도를 가진다. 이 타일드 디스플레이 응용 프로그램으로써 초고해상도 2D 맵 뷰어, 3D 렌더러, 3D Surface 렌더러, 비디오 스트림 뷰어, 원격 데스크탑 등을 개발하였다.



<그림 5> UIC 의 EVL 의 LambdaVision

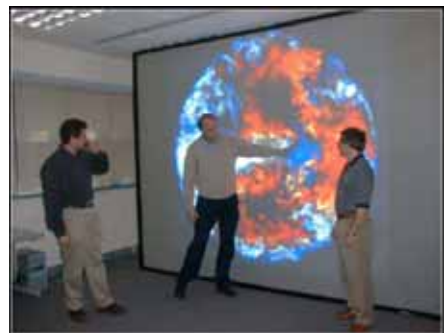
미국 프린스턴 대학[7]의 타일드 디스플레이는 6×4 의 프로젝터형 타일드 디스플레이를 구현하였다. 각각의 프로젝터 해상도는 1024×768 픽셀의 해상도를 가진다. <그림 6>과 같이 Princeton 의 타일드 디스플레이는 프로젝터 앞에 격자 사각걸이를 둬으로써 프로젝터의 빛이 사각걸이를 통과할 때 격자가 있는 부분은 빛이 감쇠되어 이웃하는 프로젝터와 빛이 겹치는

부분을 물리적인 방식에 의해 보정 작업을 수행하고 있다.



<그림 6> Princeton 대학의 타일드 디스플레이

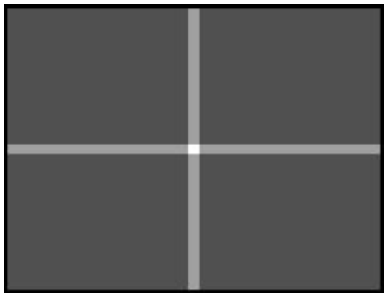
미국 NCSA(National Center for Supercomputing Applications)의 Wall-in-a-Box 는 <그림 7>과 같이 8×5 의 프로젝터형 타일드 디스플레이를 구현하였다. 투사되는 이미지의 영역이 서로 겹치지 않게 하는 Edge Align 방식을 사용하였다. 또한 Chromium[8][9]을 사용하여 OpenGL 로 구현된 프로그램을 변경없이 그대로 렌더링 할 수 있고, VNC 를 사용하여 각각의 슬레이브 컴퓨터를 하나씩 원격제어 하였으며 NCSA Pixel Blaster[10]는 마스터에서 동영상의 각 프레임 영상을 각 슬레이브로 전송하여 디스플레이하는 방식과, 로컬 컴퓨터에 디스플레이할 동영상을 사전에 배포하여 디스플레이 하는 방식이 있다. 또한 NCSA TerraServer Blaster[11]를 사용하여 이미지를 볼 수 있고, VTK Geometry Viewer [12]는 미리 계산된 Geometry 를 디스플레이 할 수 있다.



<그림 7> NCSA 의 Wall-in-a-Box

Princeton 의 타일드 디스플레이는 물리적인 방식으로 에지 블렌딩을 수행하기 때문에 시간의 변화에 따른 프로젝터의 색상변화에 대하여 쉽게

적용할 수 없다. NCSA 타일드 디스플레이 방식은 Edge Align 이기 때문에 시간의 흐름에 따라 프로젝터의 Geometry가 변화시 겹치거나 Crack 이 발생하기 쉽다. 이런 현상은 <그림 8>과 같이 눈에 잘 띠기 때문에 프로젝터간 픽셀단위로 중복없이 정확하게 재정합을 수행하여야 하므로 보정시 많은 시간이 소요된다. 그리고 일반적인 상업용 타일드 디스플레이들은 고가의 프로젝터를 사용하여 프로젝터간 영상을 겹치게 하여 프로젝터 내부의 하드웨어 기능을 사용하여 에지 블렌딩을 수행한다. <그림 8>은 4 개의 프로젝터가 서로 겹쳐서 겹치는 부분이 다른 부분보다 더 밝아지는 것을 나타낸다. 그러나 본 논문에서는 <그림 8>과 같이 이웃하는 프로젝터간의 화면을 겹치게 설정후 Programmable GPU 기법 중 Pixel Shader 기법을 사용하여 겹치는 부분에 대하여 에지 블렌딩을 수행함으로써 실시간으로 쉽게 보정이 가능하였다.



<그림 8> 이웃 프로젝터간 겹치는 장면

또한 타일드 디스플레이를 하나의 컴퓨터처럼 동작하도록 하는 타일드 디스플레이 관리 프로그램을 개발하여 사용함으로써 사용상 편리와 많은 개발 시간을 단축시킬 수 있었으며, 초고해상도의 동영상을 실시간으로 디스플레이 하기 위해서는 초고해상도 동영상을 프로젝터 수만큼 분할하여 각 슬레이브 컴퓨터에 배포하여 마스터 컴퓨터의 타임스탬프(프레임번호)를 이용하여 동기화 및 디스플레이를 수행하였다.

3D 렌더러 구현을 위해 OPENSNG 라이브러리 [13]를 사용하였다. Chromium 은 OPENGL 명령어를 스트림으로 슬레이브에 전달하여 렌더링하기 때문에 네트워크 로드가 과중되고 렌더링 속도도

늦어진다. 그러나 OPENSNG 라이브러리는 전체 장면을 Scene Graph 로 저장하기 때문에 처음 Scene Graph 를 각 슬레이브에 전송후 변경된 부분만을 전송함으로써 네트워크의 부하를 줄일 수 있어 그래픽 하드웨어에 데이터를 전달하는 방법을 최적화 할 수 있다.

또한 Scene Graph 를 통해 최종 이미지에서 쉽게 보이는 부분인지 아닌지를 체크할 수 있어서 렌더링시 성능을 크게 향상 시킬 수 있었다.

3. Tiled Display 하드웨어 구축

타일드 디스플레이 하드웨어 구성은 <그림 10,11,12>과 같이 스크린부와 프로젝터부, 클러스터부로 나누어진다.



<그림 9> IGI 의 타일드 디스플레이

스크린 부는 <그림 10>과 같이 자체 제작한 하드 스크린을 사용함으로써 소프트 스크린이 가지는 단점인 바람이나 바닥진동에 의한 흔들림을 제거할 수 있었다.



<그림 10> 스크린부

프로젝터부는 <그림 11>과 같이 프로젝터를 장착할 수 있는 프로젝터 프레임과 프로젝터를 6 자유도로 자유롭게 미세 조절할 수 있는 스테이지

및 프로젝터로 구분된다. 프로젝터 프레임은 프로젝터가 스크린에 투영시 최소의 왜곡을 가지도록 프로젝터를 M×N 으로 배치할 수 있게 설계함으로써 시간이 흐름에 따른 프로젝터의 위치 및 방향의 변화를 쉽게 보정할 수 있도록 한다. 또한 각 프로젝터는 스크린에 투영시 투영되는 부분이 직사각형을 이루고, 시리얼 통신으로 제어가 가능한 저가의 보급형 DLP 프로젝터인 OPTOMA 사의 EP-739 모델을 선정하였다.



<그림 11> 프로젝터부

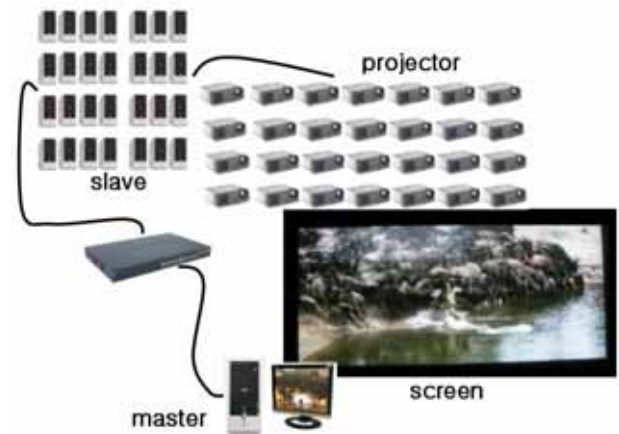
클러스터부는 <그림 12>와 같이 슬레이브와 마스터로 나눌 수가 있다.



<그림 12> 슬레이브와 마스터 컴퓨터

슬레이브는 스크린 뒤에서 프로젝션하는 프로젝터에 연결된 각각의 렌더링 클러스터 컴퓨터들을 가리키며, 마스터부는 사용자가 타일드 디스플레이를 하나의 컴퓨터처럼 운용할 수 있도록 하는 컴퓨터를 말한다. 마스터와 슬레이브만을 위한 1 기가비트 내부 네트워크로 연결되어 있으며, 마스터는 외부로부터 데이터 다운 및 인터넷 등을 위해 다른 랜카드를 통해 외부 네트워크와도 연결되어 있다. 슬레이브는 <그림 13>과 같이 프로젝터와 Geometry 를 같게 하여 1 대 1로 연결함으로써 타일드 디스플레이 운용시

발생하는 하드웨어적인 고장과 소프트웨어 에러에 대해 손쉽게 보정이 가능하도록 하였다.



<그림 13> 타일드 디스플레이 H/W 배치도

이러한 슬레이브는 각 프로젝터가 스크린의 1/(프로젝터 가로 수×프로젝터 세로 수) 부분만을 프로젝션하므로 연결된 각 슬레이브 컴퓨터는 해당 영역에 대해서만 렌더링 작업을 수행하면 된다. 이런 이유로 인해 클러스터 컴퓨터들은 저가의 보급형 컴퓨터를 사용할 수 있었다.

4. Tiled Display 소프트웨어

타일드 디스플레이의 소프트웨어로는 타일드 디스플레이 시스템을 마치 하나의 컴퓨터처럼 작동 시킬 수 있는 타일드 디스플레이 관리 프로그램, 초고해상도의 이미지 뷰어, 초고해상도 동영상 플레이어, 초고해상도 3D 뷰어로 나누어진다.

타일드 디스플레이 관리 프로그램은 <그림 14>와 같이 마스터에 타일드 디스플레이 관리 클라이언트 프로그램이 설치되고, 각 슬레이브에는 타일드 디스플레이 관리 서버 프로그램이 설치되어 부팅시 자동 로딩된다. 이 타일드 디스플레이 관리 서버 프로그램이 설치된 슬레이브는 <그림 13>과 같이 각각 시리얼 케이블로 각 프로젝트에 일대일 연결되어 있다. 타일드 디스플레이 관리 클라이언트 프로그램은 네트워크를 통해 타일드 디스플레이 관리 서버 프로그램에게 TCP/UDP 기반의 명령을 내림으로써 해당 명령을

슬레이브 컴퓨터들이 수행하게 된다. 아래와 같이 타일드 디스플레이 관리 클라이언트 프로그램의 TCP 기반의 명령은 메뉴로 등록된 명령인 프로젝터 원격 제어를 위한 프로젝터 On/Off 등과 이미지뷰어 실행 및 종료, 동영상 뷰어 실행 및 종료, 3D 뷰어 실행 및 종료 및 컴퓨터의 원격 제어인 컴퓨터 종료, 컴퓨터 재시작, 타일드 디스플레이 관리 서버 프로그램의 시작과 종료를 수행할 수 있다. 아래는 <그림 15>와 같이 타일드 디스플레이 종료를 수행하기 위해 메뉴로 등록된 10 번을 실행시키기 위해서 사용되는 명령이다.

```
daemon_client.exe - n 10
```

그러나 아래와 같이 타일드 디스플레이 클라이언트 프로그램의 UDP 명령 기반은 타일드 디스플레이의 응용프로그램이 점차 많아짐에 따라 메뉴로 등록되지 않은 테스트용으로 사용되는 임의의 프로그램들을 실행시키기 위해서 사용하였다. 아래는 마스터 컴퓨터의 이미지 뷰어 클라이언트 프로그램을 실행시킨 후 각 슬레이브의 이미지뷰어를 실행시키기 위해서 사용되는 명령이다.

```
daemon_client.exe - e pictureViewer.exe
```

타일드 디스플레이 관리 클라이언트 프로그램과 타일드 디스플레이 관리 서버 프로그램간 프로토콜은 다음과 같이 정의된다.

```
ClientIPAddress - n 1
```

```
ClientIPAddress - e pictureView.exe
```

타일드 디스플레이 관리 서버 프로그램은 자기가 속한 그룹인지를 클라이언트의 IP Address 를 통해 식별하고 해당 그룹이면 네트워크를 통해 전달된 명령 인자들을 파싱하여 - n 이면 다음 인자로 전달된 번호에 해당하는 메뉴 함수를 수행하고 - e 면 다음 인자로 전달된 프로그램을 실행한다.



<그림 14> 타일드 디스플레이 관리 클라이언트/서버들 프로그램



<그림 15> 타일드 디스플레이 종료

<그림 16>과 같은 이미지 뷰어는 동영상 플레이어와 비슷한 기반 기술을 사용하기 때문에 동영상 플레이어와 함께 설명하도록 하겠다.



<그림 16> 6952×2784 픽셀인 아시아 지도

동영상 플레이어는 동영상 클라이언트와 동영상 서버 프로그램으로 구분된다. 동영상 클라이언트 프로그램은 마스터 컴퓨터에 설치되고, 동영상 서버 프로그램은 각각의 슬레이브 컴퓨터에 설치되며 각 슬레이브에 미리 디스플레이할 동영상들을 배포한다. 그러나 동영상의 프레임의 해상도가 일정 이상 되면 각 슬레이브 컴퓨터에서 정해진 초당 프레임수를 디스플레이 할 수 없기 때문에 <그림 17>과 같이 디스플레이할 타일의 가로와 세로의 개수와 오버랩 되는 것을 고려하여 동영상을 분할하는 작업이 추가로 필요하다.



<그림 17> 7×4 동영상 분할

타일드 디스플레이에서 동영상을 보기 위해서는 프로젝터 동영상 클라이언트를 실행시킨 후 타일드 디스플레이 관리 클라이언트 프로그램을 통해서 각 슬레이브에 설치된 동영상 서버들을 일괄적으로 실행시킨다. 동영상 클라이언트는 디스플레이할 타일과 연결된 슬레이브 컴퓨터들이 마스터에 모두 로그인되면 동영상 클라이언트는 UDP 를 통해 타임스태프 (프레임 번호)를 동영상 서버에게 전달하여 해당 프레임을 화면에 디스플레이함으로써 동기화 작업을 수행할 수 있다. 그러나 타일드 디스플레이가 하나의 화면처럼 보이기 위해 프로젝터간의 겹치는 부분의 밝아진 색상에 대한 보정 작업이 필요로 한다. 색상 보정 작업을 위해서 각 슬레이브 컴퓨터는 <그림 18>과 같이 디스플레이할 비트맵과 보정용 맵들(겹치는 부분이 가장자리로 갈수록 흐려지도록 하는 맵과 프로젝터 칼라 보정 함수 맵)을 멀티텍스처를 이용하여 Pixel Shader 에서 실시간으로 칼라 보정 작업을 수행하였다.



<그림 18> 렌더링 비트맵과 보정용 맵

이런 기법은 시간 흐름에 따라 램프의 변화로 인해 겹치는 부분의 색상이 예전과 다르게 되면 프로젝터 칼라보정 함수맵만을 수정후 교체함으로써 쉽게 에지 블렌딩을 수행할 수 있다.

<그림 19>는 해상도가 1920×1080 인 HD 급 영상을 타일드 디스플레이에 디스플레이하는 것을 보여주고 있다.



<그림 19> HD 동영상

<그림 20>는 해상도가 3555×2000 픽셀인 배틀댄서의 동영상을 디스플레이하는 것을 보여주고 있다.



<그림 20> Battle Dancer 동영상

마지막으로 3D 뷰어는 OPENSF 라이브러리의 렌더링 클러스터 기법과 멀티 프로젝션을 이용하여 타일드 디스플레이를 구성할 수 있었다. 또한 에지 블렌딩은 동영상과 같은 기법을 사용하여 구현하였다. <그림 21,22>는 112 만 폴리곤을 가지는 자동차 모델을 6952×2784 픽셀로 실시간으로 렌더링 및 인터렉션하는 것을 보여주고 있다.



<그림 21> 자동차 모델 외부



<그림 22> 자동차 모델 내부

5. 결론 및 향후 계획

본 논문에서는 저가의 보급형 프로젝터를 사용하며 프로그램에서 Programmable GPU 기법인 Pixel Shader 기술을 이용하여 에지블렌딩을 수행하였으며, 7×4 개의 프로젝터로 구성된 6592×2784 픽셀의 초고해상도를 가지는 5.6m×2.4m 의 대형 타일드 디스플레이를 구축해 보았다. 또한 타일드 디스플레이 응용 프로그램으로서 타일드 디스플레이 시스템을 마치 하나의 컴퓨터처럼 작동 시킬 수 있는 타일드 디스플레이 관리 프로그램과 일반 컴퓨터에서는 실행이 불가능한 초고해상도의 이미지 및 동영상까지도 볼 수 있는 이미지 뷰어와 동영상 플레이어를 개발하였다. 또한 100 만 폴리곤 이상의 3D 모델을 실시간으로 인터랙션 할 수 있는 3D 뷰어 등을 개발 하였다. 그러나 프로젝터형 타일드 디스플레이는 LCD 형 타일드 디스플레이에 비해 이음이 보이지 않은 장점은 존재하나, 칼라매칭과 Geometric 정렬이 LCD 형 타일드 디스플레이에 비해 쉽지 않고 시간이 지남에 따라서 램프 색상의 변화와 선 및 중력으로 인한 프로젝터 Geometry 의 변화 등의 문제가 존재하며, 프로젝터와 스크린과의 투사거리 때문에 일정한 공간이 필요하다. 또한 LCD 판넬에 비해 스크린과 프로젝터는 고가이며 LCD 판넬의 수명 역시 프로젝터의 수명보다 10 배 정도 더 길다. 이러한 문제점들은 향후 개발을 통해서 개선해 나가야 할 것이다.

참고문헌

- [1]<http://www.nvidia.com/>
 [2]<http://wiki.kldp.org/wiki.php/TiledDisplay>

- [3]<ftp://grmanet.sogang.ac.kr/pub/ihm/out/041202GFK-final.pdf>
 [4]<http://dmx.sourceforge.net>
 [5]<http://ganguia.sourceforge.net>
 [6]http://www.cs.uiuc.edu/events/expwork-2005/Luc_Renambot_Abstract.pdf
 [7]<http://www.cs.princeton.edu/omnimedia/index.html>
 [8] G. Humphreys, M. Houston, Y.-R. Ng, R. Frank, S. Ahern, P. Kirchner, and J. T. Klosowski, "Chromium: A Stream Processing Framework for Interactive Graphics on Clusters," presented at SIGGRAPH, San Antonio, Texas, 2002.
 [9] Greg Humphreys, Mike Houston, Ren Ng, Randall Frank, Sean Ahern, Peter D. Kirchner, James T. Klosowski: Chromium: a stream-processing framework for interactive rendering on clusters. SIGGRAPH 2002: 693-702
 [10]<http://brighton.ncsa.uiuc.edu/~prajlich/wa11/npb.html>
 [11]<http://brighton.ncsa.uiuc.edu/~prajlich/wa11/tsb.html>
 [12]<http://www.vtk.org/>
 [13]www.opensg.org

감사의 글

"본 논문은 정보통신 선도기반기술개발사업(정보통신연구진흥원 4300-1000-1823), [실감형 가상공학 기술개발] 과제의 일환으로 수행되었음."