

# 다중 타일드 디스플레이 간의 협업 가시화 프레임워크 연구

김석환\*, 김민영\*\*, 박희찬\*\*\*, 조용주\*\*\*\*, 박경신\*\*\*  
\*츠쿠바대학교 시스템정보공학연구과 컴퓨터과학전공  
\*\*상명대학교 대학원 컴퓨터과학과  
\*\*\*단국대학교 멀티미디어공학과  
\*\*\*\*상명대학교 디지털미디어학부  
e-mail : kpark@dankook.ac.kr

## A Study on Collaborative Visualization Framework for Multiple Tiled Displays

Seokhwan Kim\*, Minyoung Kim\*\*, Heechan Park\*\*\*, Yongjoo Cho\*\*\*\*, Kyoung Shin Park\*\*\*  
\*Dept. of Computer Science, System and Information Engineering, University of Tsukuba  
\*\* Dept. of Computer Science, Sangmyung University  
\*\*\*Multimedia Engineering Major, Dankook University  
\*\*\*\*Division of Digital Media Technology, Sangmyung University

### 요 약

현재 대형 디스플레이는 여러 공공장소에 설치되어 장소의 목적에 맞는 정보를 제공한다. 이러한 대형 디스플레이는 가까운 미래에 더욱 많은 장소에 비치되어 사용될 것으로 기대된다. 또한 사용자와의 인터랙션을 통해 개인화된 정보를 제공하거나 원격지에 위치한 디스플레이 간의 상호작용도 가능할 것으로 보인다. 최근 대형 디스플레이로 고해상도의 타일드 디스플레이가 관심을 끌고 있다. 그러나 타일드 디스플레이는 분산 환경 시스템을 사용하므로 소프트웨어 개발의 복잡도가 높다. 본 논문에서는 분산환경의 타일드 디스플레이의 응용프로그램과 타일드 디스플레이들 간의 사용자 인터랙션을 통한 협업을 지원하는 확장된 iTILE 프레임워크를 살펴보고, 시스템 구조와 실험 결과를 분석한다.

### 1. 서론

최근 LCD 패널의 가격 하락과 소비자들의 고화질 이미지 선호도가 증가함으로써 공항, 지하철, 병원, 은행, 백화점 등 우리주변에서 디지털 콘텐츠를 제공하는 대형 디스플레이 사용이 엄청나게 증가하고 있다. 특히 LCD 나 PDP 를 이용한 고해상도 멀티비전 방식의 대형 타일드 디스플레이는 넓은 화면에 보다 많은 정보를 한꺼번에 보여줄 수 있어서 각종 기관의 통제 센터, 중앙감시 제어실, 방재 센터, 금융기관 등에서 적극 사용되고 있다. 따라서 현재 정보 디스플레이 제어를 지원하는 솔루션 업체도 빠르게 성장하고 있는 추세이나, 이러한 대형 디스플레이에서 기존의 마우스와 키보드를 사용하기 어려운 점으로 인하여 아직까지는 사용자 인터랙션을 고려하지 않고 주로 공공 대중을 위한 비디오 영상의 일방적인 정보 전달에만 그치고 있다.

1990 년대 이후, 컴퓨터 그래픽스 분야에서는 빔 프로젝터 기반의 대형 디스플레이를 이용한 가시화 연구가 많이 진행되었다. 그 예로 분산 디스플레이 시스템을 사용하는 CAVELib[1]나 VR Juggler[2]와 같은 가상현실 시스템 라이브러리아, 여러 개의 디스플레이를 타일과 같이 단으로 쌓아 구축하고 분산 렌더링 시스템을 구현한 Chromium[3], GARUDA[4] 등이 있다. 타일드 디스플레이는 주로 고해상도의 영상을 필요로 하는 과학적 가시화 그룹 (예: 지질학자)에서 연구용으로 사용되기 시작하였는데 한꺼번에 많은 양의 자료를 가시화하여 전체적으로 보여줄 수 있으며 가까이 다가가면 상당히 세밀한 부분까지도 자세하게 살펴볼 수도 있는 시스템을 비교적 저렴한 비용으로 구축할 수 있다는 장점이 있다.

고화질 타일드 디스플레이는 현재 여러 대의 빔 프로젝터나 LCD 를 사용하여 구성한다. 프로젝터 방식은 에지 블렌딩(Edge Blending) 기법을 사용하여 경계면이 보이지 않는 자연스러운 화면의 디스플레이를 구축할 수 있으나, 가격이 고가이며 일반적인 프로젝터의 해상도가 높지 않고, 에지 블렌딩 기능을 이용하며 해상도가 줄어들고, 프로젝터 간의 완벽한 정렬

---

이 연구는 한국과학기술정보연구원이 수행하는 기초기술연구회 협동연구사업의 일환으로 이루어졌습니다.



(그림 1) iTILE 프레임워크 구조

과 색깔 톤을 맞추기 어려운 점이 있다. 반면 LCD 방식은 모니터의 베젤(Bezel)이 이질감을 부각시켜 자연스러운 화면을 만들어 내는데 장애가 되긴 하나 가격이 저렴하고 고해상도의 디스플레이 구축이 가능하며, 하드웨어 관리가 편한 이유로 요즘 더욱 선호되고 있다. 그러나 타일드 디스플레이에서의 사용자 인터랙션은 기존의 데스크탑 컴퓨터의 WIMP (Windows Icons Mouse Point)를 확장해 적용하기 어려운 문제로 인하여 상대적으로 연구가 미약한 편이다.

대형 타일드 디스플레이는 앞으로 여러 사용자들의 인터랙션이 가능하고 개인화된 서비스를 제공하거나 원격지의 다수의 타일드 디스플레이를 연동하는 응용역시 가능할 것으로 예상된다. 예를 들어, 자동차 엔지니어와 디자이너가 타일드 디스플레이를 사용하여 디자인 회의를 하거나, 원격리에 있는 과학자들이 각자의 타일드 디스플레이를 보면서 협력적으로 가시화 분석을 하는 것이 가능하다. 이러한 경우 각각의 타일드 디스플레이에서 협력적인 가시화 응용프로그램의 실행 및 서로의 데이터를 공유하여 여러 사용자들의 인터랙션 결과가 공유되는 기능이 필요하다.

본 논문에서는 이러한 요구사항을 반영하여, 타일드 디스플레이 간의 협력 가시화를 도와주는 확장된 iTILE 프레임워크의 설계와 구현을 설명한다. 먼저 관련연구를 살펴보고 iTILE 프레임워크의 확장된 모듈들을 살펴보고, iTILE 프레임워크에 기반하여 개발된 응용프로그램들의 실행결과를 논한다.

## 2. 관련 연구

타일드 디스플레이는 일반적으로 여러 개의 컴퓨터가 각각의 디스플레이를 담당해서 처리하는 클러스터 방식의 분산 시스템을 사용하기 때문에 소프트웨어 개발이 쉽지 않다. 클러스터 기반의 타일드 디스플레이를 위한 응용프로그램을 구현하기 위해서는 네트워크를 통한 메시지 교환, 타일드 디스플레이를 구성하는 모니터와 컴퓨터의 구성 정보 분석 등 여러 가지 기능이 필요하다. 이를 위해 SAGE[5], Chromium, GARUDA 등 다양한 타일드 디스플레이를 위한 프레임워크가 개발되었다. 그러나 기존의 연구는 대부분 사용자 인터랙션을 고려한 설계보다는 분산 렌더링의 효율성에 중점을 두고 있다.

SAGE는 다른 컴퓨터에서 수행시킨 프로그램의 화면을 초고속 네트워크를 활용해서 스트리밍해서 보여주고 간단한 형태의 인터랙션을 지원하기 위해 개발되었다. Chromium은 렌더링 부하를 여러 대의 컴퓨터

에 나누어서 작업을 분산 시키는 분산 렌더링 프레임워크로 개발되었다. 하지만, 이 프레임워크는 분산 렌더링에 중심에 두고 개발되어서 타일 디스플레이에서 인터랙티브한 실시간 응용프로그램을 개발하기는 어렵다. GARUDA 같은 경우에는 Open Scene Graph (이하 OSG)로 개발된 응용프로그램을 수정없이 타일드 디스플레이에서 실행할 수 있도록 도와준다. 하지만, 마우스나 키보드 외의 사용자 인터랙션 방법을 적극적으로 지원하지 않는다.

## 3. 협력적 가시화를 위한 iTILE 프레임워크 확장

### 3.1 확장된 iTILE 프레임워크 구조

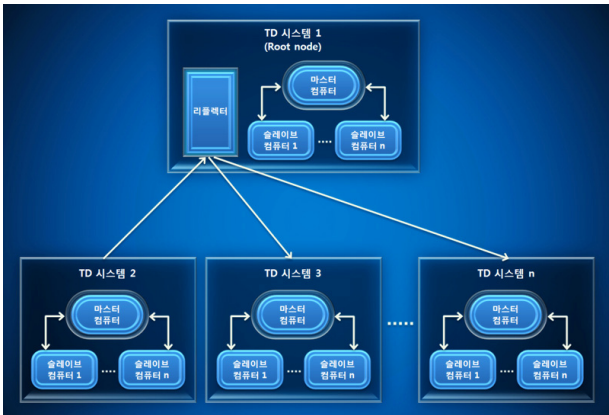
초기 iTILE 프레임워크는 타일드 디스플레이 시스템 기반에서 사용자 인터랙션을 지원하는 3차원 그래픽 응용프로그램 개발을 지원하는 것을 목적으로 개발됐다. 그림 1은 iTILE의 전체 구조를 보여준다. iTILE은 크게 마스터모듈, 슬레이브 모듈로 이루어진다.

마스터 모듈의 입력 프로세서는 입력 장치로부터 값을 읽고 그에 상응하는 인터랙션을 수행한다. 윈도우 매니저는 타일드 디스플레이 시스템에서 동작하는 응용프로그램 윈도우의 위치와 크기 같은 외관 정보를 관리한다. 사용자가 윈도우를 움직일 때마다, 각 슬레이브 노드가 그려야 하는 윈도우의 크기와 영역을 다시 계산해서 렌더링 노드에 전달한다. 입력 프로세서, 윈도우 매니저는 수행한 결과를 메시지 형태로 변환하여 이를 메시지 디스패처에 전달한다. 메시지 디스패처는 이를 네트워크를 통해 슬레이브 컴퓨터에 전달한다. 이에 따라 슬레이브 컴퓨터는 메시지를 해석하여 그에 상응하는 연산을 수행한다.

그러나 초기 iTILE 프레임워크는 타일드 디스플레이에서의 사용자 인터랙션 지원하는 여러 개의 응용프로그램의 실행에 중점을 두고, 원격지에 있는 타일드 디스플레이 간의 연동이나 여러 개의 타일드 디스플레이를 확장적으로 연결하는 부분은 고려되지 않았었다. 본 연구에서는 사용자들의 협력적 가시화 분석을 도와주기 위하여, 리플렉터(Reflector)를 이용하여 타일드 디스플레이들 간의 연동을 가능하게 하는 확장 모듈들을 추가 개발했다.

### 3.2 가상 카메라 정보 동기화 확장 모듈

타일드 디스플레이가 3차원 영상을 마치 하나의 컴퓨터가 보여주는 것처럼 렌더링하기 위해서는 각 컴퓨터 노드에서 카메라 정보를 공유하고 뷰포트만을 달리해야 한다. 따라서 iTILE을 이용해 타일드 디스플레이 슬레이브 노드들은 3차원 영상을 렌더링하기 위해 필요한 가상 카메라 정보를 자동으로 공유한다. 그러므로 여러 개의 타일드 디스플레이 시스템들이 함께 작동하는 협력적 가시화를 위해서는 이 카메라 정보가 공유되어야 한다. 하지만, 기존의 iTILE은 하나의 타일드 디스플레이 시스템만을 대상으로 제작되었기 때문에 이를 지원하는 모듈을 갖고 있지 않았다.



(그림 2) 여러 개의 타일드 디스플레이들 간의 연결을 위한 Reflector 구조

본 연구에서는 이점을 해결하고자 가상 카메라 정보를 동기화된 방법으로 공유하는 확장 모듈을 개발했다. 이 확장 모듈은 그림 2에 보이는 것과 같이 일종의 리플렉터 (Reflector)로 각 타일드 디스플레이 시스템과 리플렉터 간에 TCP 네트워크를 유지한다. 그렇게 되면 어떤 타일드 디스플레이 시스템에서 카메라 정보를 리플렉터에 전달하면 이 리플렉터는 연결된 다른 모든 타일드 디스플레이 시스템들에게 이 정보를 전달한다.

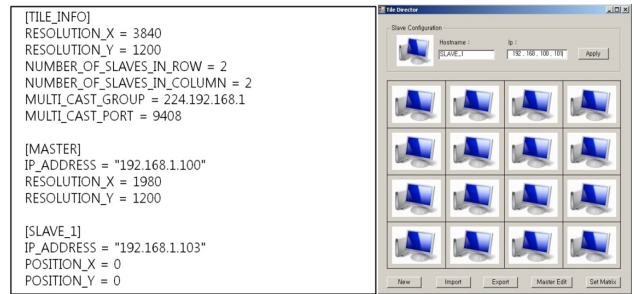
한 프레임을 렌더링하는 과정을 살펴보면, 먼저 최초 인터랙션을 수행하는 타일드 디스플레이 시스템의 마스터 컴퓨터는 변경된 가상 카메라 정보를 자신이 담당하는 슬레이브 컴퓨터들과 리플렉터에 동시에 전송한다. 그 후, 리플렉터는 이 정보를 다시 연결된 모든 타일드 디스플레이 시스템의 마스터 컴퓨터에 전달하고, 각각의 타일드 디스플레이의 마스터 컴퓨터는 이를 다시 자신이 담당하는 슬레이브 컴퓨터에 전달하는 순서로 진행된다.

### 3.3 분산 공유 메모리 확장 모듈

협력적 가시화를 위하여 iTILE 프레임워크에서는 분산 공유 메모리 확장 모듈을 개발했다. 분산 공유 메모리는 각 슬레이브 노드에서 실행되는 iTILE 응용 프로그램 간에 같은 렌더링 결과를 보여주기 위하여 동적으로 변경되는 값을 공유하여 유지시켜준다. iTILE의 분산 공유 메모리는 신뢰성 있는 멀티캐스트 프로토콜인 PGM[6]을 활용하여 경량 분산 공유 메모리로 제작됐다.

분산 공유 메모리는 사용자의 입력 없이 개체들의 동적인 움직임이 가능하도록 한다. 예를 들어, 잠자리들이 날아다니는 모습을 렌더링하는 경우 각 슬레이브 노드에서 잠자리들이 언제 어느 위치에 날아다니는 지에 대한 시뮬레이션 데이터를 공유 메모리 모듈을 통해서 렌더링 동기화에 반영하게 된다.

타일드 디스플레이 기반의 응용 프로그램에서 동적으로 움직이는 개체가 존재할 경우 이런 개체들의 정보는 협력적 가시화를 위해 연결된 모든 타일드 디스플레이들 간에 공유되어야 한다. 이를 위해서 분산 공유 메모리 데이터는 2.2 절에 설명한 리플렉터를 활



(그림 3) 설정 파일 예제 (왼쪽) GUI를 활용한 설정파일 편집기 (오른쪽)

용하여 각 타일드 디스플레이 시스템들 간에 공유될 수 있도록 했다.

### 3.4 설정파일 GUI 편집기

현재 iTILE 응용프로그램의 실행을 위하여 그림 3의 왼쪽에 보이는 것과 같은 설정파일을 이용한다. 이 설정 파일에는 각 타일드 디스플레이 시스템의 구성정보 (모니터의 가로 x 세로 개수와 해상도 및 위치 정보), 네트워크 통신에 필요한 각 컴퓨터의 IP 주소 정보를 가지고 있다. 그런데 하나의 타일드 디스플레이를 기반으로 하는 iTILE 응용 프로그램의 경우는 하나의 타일드 디스플레이의 설정파일만을 편집하여 실행하여 사용하면 된다. 또한 설정파일은 일반 텍스트 편집기를 사용하여 직접 수정이 가능하며, 그림 3의 오른쪽에 보이는 GUI 기반의 설정파일 편집기를 사용할 수도 있다.

그리고 여러 개의 타일드 디스플레이를 연결하는 경우, 각 타일드 디스플레이 시스템에 대한 설정 파일을 가져야 하며, 각 타일드 디스플레이 노드마다 응용프로그램의 자동 설치 및, 한 번의 응용프로그램 실행으로 연결되어 있는 모든 타일드 디스플레이들의 노드에서 실행이 가능하도록 하는 부분은 현재 GUI 기반의 설정파일 편집기 프로그램에 추가하도록 개발하는 중이다.

## 4. 실행 결과

그림 4는 확장된 iTILE 프레임워크를 기반으로 제작된 간단한 협력적 가시화 응용프로그램의 실행 모습이다. 이 프로그램에서 4 X 3으로 구성된 타일드 디스플레이 시스템과 2X2로 구성된 테이블 탑 시스템을 연동하였다. 그림 4 위는 3차원 모델 뷰어 프로그램으로, Open Scene Graph (OSG)에서 기본적으로 제공하는 모델 파일인 Cow 모델을 띄우고 이를 마우스를 활용해 간단한 인터랙션을 수행할 수 있다. 프로그램 실행 결과 타일드 디스플레이 시스템에 연결된 마스터 컴퓨터에서 마우스로 모델을 회전하는 경우 Cow 모델은 테이블탑에서도 동기화되어 실시간으로 회전하는 것을 확인할 수 있었다.

그림 4 오른쪽은 정육면체 내부에 나비가 위치하고 있고, 나비가 난수를 활용하여 계속해서 방향과 위치를 바꾸며 움직이는 응용 프로그램이다. 이 응용 프



(그림 4) 두 개의 타일드 디스플레이 간의 연동프로그램 Cow Model Viewer (왼쪽)와 Dynamic Object Animation (오른쪽)

로그래머에서 나비는 사용자의 입력 없이 내부적으로 난수를 활용하여 동적으로 움직이기 때문에, 분산 공유 메모리와 리플렉터 간의 동작을 확인하기 위해 제작되었다. 실험 결과 이 역시 실시간으로 타일드 디스플레이와 테이블탑 간에 동기화되어 나비 개체가 움직이는 것을 확인할 수 있었다.

## 5. 결론 및 향후 연구 방향

미래에는 도처에 고해상도 대형 디스플레이가 설치되어 사용자들과 상호작용을 통해 유익한 정보를 제공할 것이라고 예측된다. 하지만, 현재 단일 디스플레이로서 충분한 고해상도 대형 디스플레이가 존재하지 않기 때문에 이를 위한 프로토타입으로 타일드 디스플레이가 개발되었다. 하지만 타일드 디스플레이는 여러 개의 분산 컴퓨터 시스템에서 운용하기 때문에 소프트웨어 개발의 복잡도가 높다. 따라서 이를 위한 iTILE 이라는 타일드 디스플레이 응용프로그램 개발 프레임워크가 개발되었다. 본 연구에서는 iTILE의 특징을 간략히 살펴보고 기존의 iTILE을 활용하여 다수의 타일드 디스플레이간의 협업 환경을 구성할 수 있는 iTILE의 확장에 대해 살펴보았다. 하지만, 아직 다수의 타일드 디스플레이를 활용하는 iTILE의 확장은 몇 가지 문제점이 있다.

두 개의 타일드 디스플레이 시스템을 연동시켜 실험해보았을 때, 가상 카메라 정보를 공유 부분 즉 렌더링을 동기화시키는 부분은 초당 30 프레임 이상을 보이고 있어, 실시간으로 응용프로그램을 실행시키는데 지장이 없었다. 하지만, PGM 기반의 분산 공유 메모리 확장 모듈은 타일드 디스플레이 시스템 간에 연동시켰을 때에는, 초당 약 15 번 정도 동기화 시키는 것으로 나타났다. 이는 UDP 기반의 멀티캐스트를 활용하던 화면 렌더링에 비해 떨어지는 성능이므로, 향후 연구에서는 데이터 공유 동기화 성능을 좀 더 최적화 시킬 계획이다.

본 논문에서 활용한 모듈은 현재는 iTILE 프레임워크를 활용한 확장 모듈의 형태로 존재하는데, 향후 연구에서는 이를 프레임워크 수준에서 지원할 수 있도록 모듈을 프레임워크 내부로 삽입할 것이다. 그리

고 이를 활용하여 원거리 타일드 디스플레이 시스템 간의 인터랙션 활용 시나리오를 개발하고 응용 프로그램을 개발하여 사용성 평가를 진행할 것이다.

## 참고문헌

- [1] VRCO, CAVELib. <http://www.vrco.com>
- [2] VRJuggler, <http://www.vrjuggler.org>
- [3] Greg Humphreys, Mike Houston, Ren Ng, Randall Frank, Sean Ahern, Peter Kirchner, James T. Klosowski, "Chromium: a stream-processing framework for interactive rendering on clusters," ACM Transactions on Graphics, Vol. 21, No. 3, pp. 693-702, 2002.
- [4] Nirimesh, Pawan Harish, P. J. Narayanan, "Garuda: A Scalable, Tiled Display Wall Using Commodity PCs," IEEE Transactions on Visualization and Computer Graphics, Vol. 13, No. 5, Sept/Oct, 2007.
- [5] Byungil Jeong, Luc Renambot, Ratko Jagodic, Rajvikram Singh, Julieta Aguilera, Andrew Johnson, Jason Leigh, "High-Performance Dynamic Graphics Streaming for Scalable Adaptive Graphics Environment," in proceedings of IEEE Information Visualization Workshop, 2005.
- [6] Anthony Jones, Jim Ohlund, "Network Programming for Microsoft Windows," Microsoft, 2002.